Saket Sharma

EMRS TGT LIBRARIAN 2025 TIER 2-LIS

Topic Covered
Part 3 - All Units

Copyright © 2025 – Saket Sharma

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means—electronic, mechanical, photocopying, recording, scanning, or otherwise—without the prior written permission of the author/publisher, except as permitted under Section 52 of the Indian Copyright Act, 1957, for purposes of fair use such as review, research, criticism, or academic reference.

This book has been prepared exclusively for academic and educational purposes, based on the official **EMRS TGT Librarian Tier 2 Part 3 syllabus**. While every effort has been made to ensure accuracy and organization, the syllabus and related materials remain the intellectual property of the **National Education Society for Tribal Students (NESTS)**, Government of India.

Disclaimer

This publication is an independent academic resource designed to assist candidates preparing for **EMRS TGT Librarian 2025** and other Library and Information Science (LIS) examinations. It is *not* an official publication of NESTS or the Ministry of Education. Readers are advised to refer to official notifications and documents for authoritative information.

License

This work is licensed under a **Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License (CC BY-NC-SA 4.0).** You are free to share and adapt the material for **non-commercial purposes**, provided appropriate credit is given and any derivative works are distributed under the same license. **Commercial use is strictly prohibited** without prior written permission from the author.

Author & Designer: Saket Sharma

Edition: First Edition

Year of Publication: 2025

Release Date: 24 October 2025

Printed & Published by: Saket Sharma

Place of Publication: New Delhi, India

To Connect

Website: www.saketsharma.com

Telegram Channel: https://t.me/UGCNETJRF2024JUNE

LinkedIn: https://www.linkedin.com/in/saket-sharma-3362b7b5/

WhatsApp Channel: https://whatsapp.com/channel/0029Vatnly48KMqf9x3wbp04

MCQ Portal: https://www.saketsharma.com/meg-portal

Contents

Unit 1 – Information Technology 4

Unit : 2 Computer Basic 7

Unit : 3 Computer Architecture-Organization of Computer 18

Unit: 4 Software 23

Unit. 5 MS-Office packages 42

Unit : 6 DBMS Package 46

Unit : 7 Computer application to library & Information work 57

Unit : 8 Communication Technology 63

EMRS TGT Part 3 -Librarian Syllabus

Part III: Information Technology – Basic

Unit No.	Title	Contents / Topics
Unit 1	**Information Technology**	• Definition • Need • Scope • Objectives
Unit 2	**Computer Basics**	• Introduction to Computers • Overview of Historical Development of Computers • Generations of Computers • Classification of Computers • Essential Components of a Computer System
Unit 3	**Computer Architecture – Organization of Computer**	• Input and Output Devices: – Keyboard – Scanner – OCR – Printers – Monitor
Unit 4	**Software**	• Operating Systems: – Single & Multi User Systems – MS-DOS, MS Windows, Linux, UNIX, Windows NT (Basic Features) • Programming Languages: Concepts and Tools • Algorithm & Flowcharting
Unit 5	**MS-Office Package**	• Overview and basic functions of MS Word, MS Excel, MS PowerPoint, and MS Access
Unit 6	**DBMS Package**	• Familiarity with: – DBASE – FOXPRO – CDS/ISIS – SOUL – MS Access (Basic features)
Unit 7	**Computer Application to Library & Information Work**	• Housekeeping Operations (e.g., cataloguing, circulation, acquisition)
Unit 8	**Communication Technology**	• Communication Technology – Basic Concepts • Networking – Basic Concepts • Internet – Definition, Uses, and Components

Unit 1 – Information Technology

Information Technology (IT)

1. Definition:
Information Technology (IT) refers to the **study, design, development, implementation, and management of computer-based information systems**, particularly software applications and computer hardware.
It involves the **use of computers, communication networks, and electronic systems** to store, retrieve, process, and transmit data in various forms such as text, images, and sound.

IT encompasses everything from basic data entry and storage to complex technologies like artificial intelligence and cloud computing.

Examples:

- Using a **computer and word processing software** (like MS Word) to prepare reports or letters.
- Managing **library databases** for cataloguing and circulation.
- Sending and receiving **emails** or sharing files via the **Internet**.
- Using **digital reference services** in libraries for remote user assistance.

2. Need:
The **need for IT** arises from the exponential growth of information and the increasing requirement for **speed, accuracy, and efficiency** in managing and disseminating it.
In libraries, educational institutions, and organizations, manual systems cannot handle today's information load effectively — hence the adoption of IT tools becomes essential.

Key Reasons for IT Need:

1. **Information Explosion:**
   Managing vast quantities of data efficiently requires automated systems.
2. **Quick Communication:**
   Enables instant global communication via email, video conferencing, and messaging.
3. **Accuracy and Reliability:**
   Reduces human error in data processing and information retrieval.
4. **Resource Sharing:**
   Makes it possible to share resources across different locations and institutions (e.g., through library networks like DELNET or INFLIBNET).
5. **User Convenience:**
   Allows users to access data and information remotely at any time.

Examples:

- Providing **instant access** to digital journals and e-books in academic libraries.
- Using **video conferencing tools** for online meetings or classes.
- Automating **acquisition, cataloguing, and circulation** in libraries to save time and improve accuracy.

3. Scope:
The **scope of IT** is vast and continues to expand with technological progress.
It impacts **almost every aspect of human activity** — from education and business to healthcare, government, and social life.

In the field of **Library and Information Science (LIS)**, IT revolutionizes traditional library operations by automating manual functions, supporting digital resources, and enabling networked information sharing.

Main Areas of IT Application:

1. **Education:**
   Online learning platforms, e-books, and virtual classrooms.
2. **Libraries and Information Centers:**
   Library automation, digital catalogues (OPAC), institutional repositories, and e-resources.
3. **Business and Commerce:**
   Online transactions, e-commerce websites, data analytics, and enterprise management systems.
4. **Health Sector:**
   Electronic medical records, telemedicine, and data analysis for research.
5. **Government Administration:**
   E-governance, public information portals, and digital record management.
6. **Research and Development:**
   Use of databases, simulation tools, and data visualization software.

Examples:

- **Digital libraries** providing access to thousands of e-resources.
- **Cloud computing** for storing and accessing research data securely.
- **Online Public Access Catalogue (OPAC)** for locating library materials.
- Use of **RFID technology** in libraries for tracking and issuing books.

4. Objectives:
The **main objectives** of implementing Information Technology are to enhance efficiency, accuracy, accessibility, and connectivity in information handling.
It aims to **simplify complex processes**, enable **automation**, and **empower decision-making** by ensuring timely access to reliable information.

Major Objectives:

1. **Efficiency and Speed:**
   To reduce time spent on routine tasks and provide faster access to information.
2. **Accuracy and Reliability:**
   To minimize errors in data entry, processing, and retrieval.
3. **Automation:**
   To replace repetitive manual processes with computerized systems.
4. **Accessibility:**
   To make information available to users anywhere and anytime.

5. **Knowledge Sharing:**
   To facilitate global collaboration and communication.
6. **Security:**
   To protect information from loss, unauthorized access, or misuse.

Examples:

- Using **Integrated Library Management Software** like **SOUL**, **KOHA**, or **LibSys** for library automation.
- Maintaining **digital archives** for preserving rare manuscripts and research data.
- Providing **online library access** through institutional websites.
- Using **data backup systems** and **cloud servers** to safeguard library data.

Unit : 2 Computer Basic

5. Evolution of Computers

The history of computers spans over 2000 years, beginning with simple counting devices and evolving into modern digital machines.

5.1 Early Devices

- **Abacus:** Invented around 2000 years ago; used beads on rods for arithmetic calculations.
- **Slide Rule (1620):** Invented by William Oughtred; used for multiplication and division based on logarithms.
- **Pascal's Calculator (1642):** Built by Blaise Pascal; a mechanical adding machine using numbered wheels.
- **Leibnitz's Calculator (1671):** Improved machine capable of addition and multiplication using stepped gears.
- **Thomas of Colmar's Calculator (1820s):** First successful commercial mechanical calculator performing all basic operations.

5.2 The First Computer

- **Charles Babbage (1822):** Designed the *Difference Engine*, an automatic mechanical calculator. Later developed the concept of the *Analytical Engine*, considered the first design of a modern computer.
- **Herman Hollerith (1890):** Used *punched cards* for data processing in the U.S. Census; foundation of early data storage.
- The punched card technology led to the rise of companies like **IBM** and **Remington**.

5.3 Some Well-Known Early Computers

- **Mark I (1944):** Developed by *Howard Aiken* at Harvard University.
  Performed arithmetic operations automatically; multiplication took about 4 seconds, division 11 seconds.
- **ENIAC (1946):** Built by *Eckert and Mauchly* at the University of Pennsylvania.
  Used 18,000 vacuum tubes and was the first high-speed electronic digital computer.
- **EDVAC (1950):** Designed using *John Von Neumann's stored program concept*, where data and instructions were stored in memory.
- **UNIVAC (1951):** The first *commercially produced* digital computer, used by the U.S. Census Bureau.

5.4 Computer Generations

Each generation marks a major leap in technology — mainly in hardware, processing speed, and software capability.

First Generation (1940–1956):

- Used *vacuum tubes* and *magnetic drums* for memory.
- Very large, produced much heat, and consumed a lot of power.

- Programming done in *machine language*.
- Example: ENIAC, UNIVAC.

Second Generation (1956–1963):

- Used *transistors* instead of vacuum tubes — smaller, faster, and more reliable.
- Introduced *assembly language* programming.
- Example: IBM 7094, CDC 1604.

Third Generation (1964–1971):

- Used *integrated circuits (ICs)*.
- Increased processing speed and storage.
- Introduced *operating systems* for multitasking.
- Example: IBM 360 series, PDP-8.

Fourth Generation (1971–1980s):

- Used *microprocessors* (chips combining thousands of ICs).
- Personal computers (PCs) introduced.
- Examples: Apple II, IBM PC.

Fifth Generation (1980s–Present):

- Based on *Artificial Intelligence (AI)* and *parallel processing*.
- Use of *Very Large Scale Integration (VLSI)* and *Ultra Large Scale Integration (ULSI)* chips.
- Examples: Modern laptops, AI-based systems.

5.5 First Generation Computers (1940–1956)

- Used **vacuum tubes** for circuitry and **magnetic drums** for memory.
- Operated with **machine language** (binary code), allowing only one problem at a time.
- Data and instructions were entered using **punched cards** and **paper tapes**.
- Output was printed on **paper printouts**.
- Examples: **ENIAC**, **EDVAC**, **UNIVAC**.
- These computers were **huge, expensive**, and **generated a lot of heat**.

5.6 Second Generation Computers (1956–1963)

- Introduction of **transistors** and **magnetic core memory** revolutionized computing.
- Transistors made computers **smaller, faster, more reliable**, and **energy-efficient** compared to vacuum tubes.
- Programming shifted from **machine language** to **assembly language**.
- **High-level languages** such as **COBOL** (for business) and **FORTRAN** (for scientific calculations) were developed.

- Still **expensive to operate** and found mainly in **large organizations and laboratories**.
- Improved **storage capacity** and **processing speed**.

5.7 Third Generation Computers (1964–1971)

- Marked by the use of Integrated Circuits (ICs) developed by Jack Kilby in 1958.
- ICs replaced transistors, making computers smaller, cheaper, and more efficient.
- Keyboards and monitors replaced punched cards and printouts.
- Operating systems introduced, enabling multiple applications to run simultaneously.
- Widely used in business applications such as:
  • Accounting
  • Payroll
  • Billing
  • Inventory management
- These machines were more user-friendly and cost-effective.

5.8 Fourth Generation Computers (1971–Present)

- Based on microprocessors, using Large Scale Integration (LSI) and Very Large-Scale Integration (VLSI) technology.
- Thousands of components fitted onto a single silicon chip.
- Led to the rise of Personal Computers (PCs) — affordable, compact, and powerful.
- Apple and IBM were pioneers in commercial PCs.
- Memory chips developed in 1970 could store 256 bits, greatly improving storage capability.
- ICs used for both processing and memory construction.
- Marked the beginning of mass computing for homes, schools, and businesses.

5.9 Fifth Generation Computers (The Road Ahead)

- Based on Artificial Intelligence (AI) and parallel processing.
- Aim: to create machines capable of learning, reasoning, and natural language understanding.
- Utilize VLSI and Ultra-Large-Scale Integration (ULSI) technologies.
- Features include voice recognition, machine learning, and expert systems.
- Focus on automation, intelligent decision-making, and self-learning systems.
- Still evolving, forming the base for modern robotics and AI research.

6. Components of a Computer
A computer system is primarily composed of **three main units** —

1. **Input Unit**
2. **Central Processing Unit (CPU)**
3. **Output Unit**
   Additionally, it includes **Secondary (Auxiliary) Storage** for long-term data storage.

6.1 Input Unit

- Used to **enter data and instructions** into the computer for processing.
- Converts human-readable input into machine-readable form.
- Common input devices include **keyboard, mouse, scanner, microphone**, etc.
- Example: Entering student records through a keyboard for database storage.

6.2 Central Processing Unit (CPU)

- Known as the **brain of the computer**.
- Executes instructions and controls the functioning of all other components.
- Comprises three main parts:
    1. **Arithmetic Logic Unit (ALU)**
    2. **Control Unit (CU)**
    3. **Main Memory (Primary Storage)**

6.3 Arithmetic Logic Unit (ALU)

- Performs all **arithmetic operations** such as addition, subtraction, multiplication, and division.
- Handles **logical operations** like comparison — greater than, less than, or equal to.
- Example: Comparing two numbers to check which is larger.

6.4 Control Unit (CU)

- Directs the operation of all parts of the computer system.
- **Fetches, decodes, and executes** program instructions in sequence.
- Coordinates with input, output, memory, and ALU for smooth functioning.
- Example: CU sends instructions to ALU to perform addition, then directs the result to memory.

6.5 Main Memory (Primary Storage)

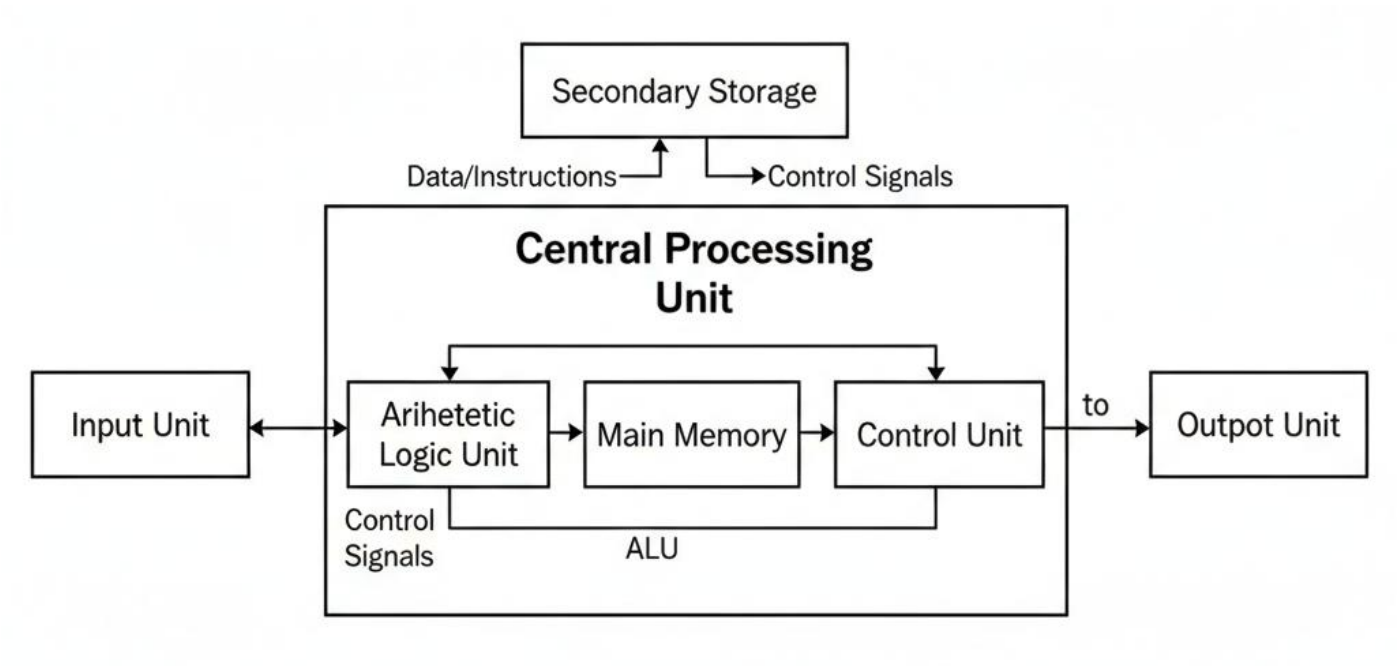
- Temporarily holds **data and instructions** while processing takes place.
- Also stores **intermediate and final results** before output.
- Data in main memory is **volatile**, meaning it is lost when power is turned off.
- Examples: **RAM (Random Access Memory)** and **ROM (Read Only Memory)**.

6.6 Output Unit

- Converts processed data from machine-readable form into human-readable form.
- Displays or prints results for the user.
- Output can be **soft copy (screen display)** or **hard copy (printed report)**.
- Examples: **Monitor, Printer, Speaker**.

6.7 Secondary Storage (Auxiliary Storage)

- Used for **permanent and large-scale data storage**.
- Cheaper and slower than main memory but **non-volatile**.
- Stores data, programs, and backups that are not in immediate use.
- Examples: **Hard Disk, CD/DVD, Pen Drive, Magnetic Tape**.



7. Hardware vs Software

- **Hardware** refers to the **physical components** of a computer that you can see and touch.
  Examples: CPU, keyboard, monitor, printer, hard disk, cables, etc.
- **Software** refers to the **set of instructions or programs** that tell the hardware what to do.
  Examples: Windows, MS Word, Photoshop, antivirus programs, etc.

Difference Between Hardware and Software

Basis	Hardware	Software
Meaning	Physical components of the computer system.	Set of programs or instructions used to operate the computer.
Nature	Tangible (can be touched).	Intangible (cannot be touched).
Dependency	Works only when software is installed.	Requires hardware to run.
Examples	CPU, Monitor, Keyboard, Printer.	Windows OS, MS Office, Tally, Photoshop.
Failure	May suffer from physical damage.	May get corrupted or deleted.

8. System Software vs Application Software

- **System Software:**
  Controls and manages computer hardware. Acts as an interface between the hardware and the user.
  Examples: **MS-DOS, UNIX, Windows, Linux, MacOS**.
- **Application Software:**
  Designed to perform **specific tasks** for users. Runs on top of system software.
  Examples: **MS Word, Excel, Tally, Database Software, Payroll Software**.

Difference Between System and Application Software

Basis	System Software	Application Software
Purpose	Manages computer hardware and system operations.	Performs specific user-oriented tasks.
Users	Used by system administrators and computers internally.	Used by end users.
Dependency	Works independently.	Depends on system software.
Examples	Windows, Linux, UNIX.	MS Word, Excel, Tally, Photoshop.

9. Bits and Bytes

- **Bit:**
  The smallest unit of data in a computer; represents either **0** or **1** (binary digits).
  0 = no electrical signal, 1 = presence of electrical signal.
- **Byte:**
  A group of **8 bits** that represents a single character (alphabet, digit, or symbol).

Common Data Units

Unit	Equivalent	Description
1 Byte	8 Bits	Stores one character (e.g., 'A', '7', '#')
1 Kilobyte (KB)	1,024 Bytes	Small text file
1 Megabyte (MB)	1,048,576 Bytes	Medium image or document
1 Gigabyte (GB)	1,073,741,824 Bytes	Video file or large program
1 Terabyte (TB)	≈1,099,511,000,000 Bytes	Hard drive capacity
1 Petabyte (PB)	≈1,125,899,900,000,000 Bytes	Data center storage

Key Points:

- All data in a computer is represented using **binary digits (0 and 1)**.
- Bytes are used to measure **memory size**, **file size**, and **data transfer capacity**.

10. Input and Output Devices

Input and Output (I/O) devices allow users to **communicate with the computer system**.

Input devices bring information *into* the computer.
Output devices send processed information *out* to the user.

Common Input and Output Devices

Input Devices	Output Devices
Keyboard	Monitor
Mouse	Printer
Joystick	Plotter
Scanner	LCD Display
Light Pen	Speaker
Touch Screen	Projector

10.1 Keyboard

- A **text-based input device** for typing alphabets, numbers, and symbols.
- Contains different sets of keys:
  - **Alphanumeric Keys:** A–Z, 0–9, and special symbols (+, –, /, *).

- **Function Keys (F1–F12):** Perform specific software-based tasks.
- **Special Keys:** Perform control operations like moving, deleting, inserting, etc.

Important Keys and Functions

Key	Function
Enter	Executes a command or confirms input.
Spacebar	Inserts space between words.
Backspace	Deletes the character to the left of the cursor.
Delete	Removes the character at the cursor position.
Insert	Switches between insert and overwrite modes.
Shift	Types uppercase letters or upper symbols.
Caps Lock	Toggles uppercase typing mode.
Tab	Moves cursor to next tab position.
Ctrl / Alt	Used in combination with other keys for shortcuts.
Esc	Cancels or exits a command.

- **Numeric Keypad:** Located on the right; used for quick numeric data entry.
- **Cursor Keys:** Arrow keys to move the cursor up, down, left, or right.

10.2 Mouse

- A **pointing device** used to control the on-screen cursor.
- Has two or more buttons; most common are **left** and **right** buttons.

Common Mouse Actions

Action	Purpose
Left Click	Select an item.
Double Click	Open a file or program.
Right Click	Display shortcut menu or commands.
Drag and Drop	Move or copy items by holding and dragging.

11. Scanner

- A **scanner** is an **input device** used to directly capture data from documents and convert it into digital form.
- It reduces data entry errors and allows quick input of printed text or images.
- Commonly used in offices, libraries, and retail stores (barcode scanners).

Types of Scanners

Type	Description / Example
Flatbed Scanner	Most common; document placed on a glass surface for scanning.
Hand-held Scanner	Used in stores to scan barcodes or item prices.
Sheet-fed Scanner	Feeds paper automatically for scanning documents in bulk.

12. Light Pen

- A **pen-shaped input device** used to select or draw directly on a computer screen.
- Works by detecting the light emitted from the display surface.
- Commonly used in **Computer-Aided Design (CAD)** and graphic applications.

13. Touch Screen

- Allows users to **interact with a computer by touching** options displayed on the screen.
- Used in **ATMs, information kiosks, smartphones**, and ticket machines.
- Combines **input and output functions** in one device.

14. Output Devices

Output devices present processed data in a human-readable form.
They are mainly of two types: **Soft Copy Output** (displayed on screen) and **Hard Copy Output** (printed on paper).

14.1 Monitor

- A **display device** resembling a television screen.
- Uses **Cathode Ray Tube (CRT)** or **Liquid Crystal Display (LCD)** technology.
- Displays both input data and processed results.

Types of Monitors

Type	Features
CRT Monitor	Bulky, high power consumption, low cost.
LCD Monitor	Slim, lightweight, energy-efficient, used in laptops and modern PCs.

14.2 Printer

- A **hard copy output device** that produces text or graphics on paper.
- Classified into two main types: **Impact** and **Non-Impact** printers.

Types of Printers

Type	Technology	Examples	Features
Impact Printers	Print by striking ribbon against paper.	Dot Matrix, Daisy Wheel	Noisy, slower, low-quality output.
Non-Impact Printers	Use chemical, electrical, or heat-based printing.	Inkjet, Laser, Thermal	Quiet, fast, and high-quality printouts.

Printer Quality Measures

Measure	Description	Units
Resolution	Number of dots printed per inch.	DPI (Dots per Inch)
Speed	Number of characters, lines, or pages printed per unit time.	CPS (Characters per Second), LPM (Lines per Minute), PPM (Pages per Minute)

14.3 Plotter

- A specialized **output device** used to produce **graphics, drawings, charts, and maps**.
- Works with **automated pens** to draw precise lines on paper.
- Commonly used in **CAD/CAM and engineering design applications**.

Types of Plotters

Type	Description
Flatbed Plotter	Paper remains stationary while pens move across it.
Drum Plotter	Paper moves around a rotating drum while pens draw.

15. RAM and ROM

Computer memory is divided into two main types — **RAM (Random Access Memory)** and **ROM (Read Only Memory)**.

15.1 RAM (Random Access Memory)

- Known as **primary or main memory**.
- Temporarily stores **data, programs, and instructions** currently in use.
- Data is **volatile**, meaning it is lost when the computer is turned off.
- Often referred to as **Read/Write Memory**.
- More RAM = faster performance.

15.2 ROM (Read Only Memory)

- **Non-volatile memory** — retains data even after power is turned off.
- Contains **manufacturer's instructions** and **bootstrap loader** (used to start the computer).
- Data in ROM is **read-only** and cannot be modified easily.

Types of ROM

Type	Full Form	Description
PROM	Programmable ROM	Can be programmed once by the user.
EPROM	Erasable Programmable ROM	Can be erased using UV light and reprogrammed.
EEPROM	Electrically Erasable Programmable ROM	Can be erased and rewritten using electrical signals.

16. Secondary Storage Devices

- **Secondary storage**, also called **auxiliary storage**, is used for **permanent data storage**.
- It supplements the limited and volatile memory of **RAM**.
- These devices are **non-volatile**, meaning data remains stored even when power is turned off.
- Examples include **Floppy Disk, Hard Disk, CD, DVD, Pen Drive, and Zip Drive**.

Key Features of Secondary Storage

- Provides **large storage capacity**.
- **Cheaper** than primary memory.
- **Portable** and **reusable**.
- Used for **backups** and **data transfer**.

16.1 Floppy Disk

- A small, **removable magnetic disk** used for portable data storage.
- Common size: **3.5 inches**, capacity **1.44 MB**.
- Can be **read and rewritten** multiple times.
- Used earlier to transfer small files between computers.

Advantages:

- Portable and inexpensive.

Disadvantages:

- **Low storage capacity** and **easily damaged**, making it unreliable for long-term use.

16.2 Hard Disk

- A **rigid magnetic disk** used as the main storage device in computers.
- Usually a stack of metal **platters** enclosed in a sealed container.
- Storage capacity ranges from **1 GB to several TB (terabytes)**.
- Much **faster and more reliable** than floppy disks.
- Can store **operating systems, applications, and large data files** permanently.

Types of Hard Disks

Type	Description
Disk Pack	Removable; can be taken offline when not in use.
Winchester Disk	Fixed and sealed inside the computer; more reliable and durable.

Advantages:

- High speed, large capacity, and long lifespan.

Disadvantages:

- Not easily portable, more expensive than floppy disks.

16.3 Compact Disc (CD)

- **Optical storage device** that uses laser technology to read and store data.
- Capacity: **650–700 MB**.
- Ideal for storing **music, videos, text, and software**.
- Data is **read-only** and cannot be rewritten unless it is a **CD-RW (rewritable)** version.
- Commonly used for **software distribution** and **media storage**.

Advantages:

- Reliable, portable, and inexpensive.

Disadvantages:

- Can be easily scratched or damaged; limited storage compared to modern devices.

16.4 Digital Versatile Disc (DVD)

- An improved version of the CD with **larger capacity** and **better quality**.
- Capacity: from **4.7 GB (single layer)** to **8.5 GB (dual layer)**.
- Used mainly for **storing movies, music, and large applications**.
- Like CDs, most DVDs are **read-only**, though rewritable types (DVD-RW) also exist.

Advantages:

- High storage capacity and superior video/audio clarity.

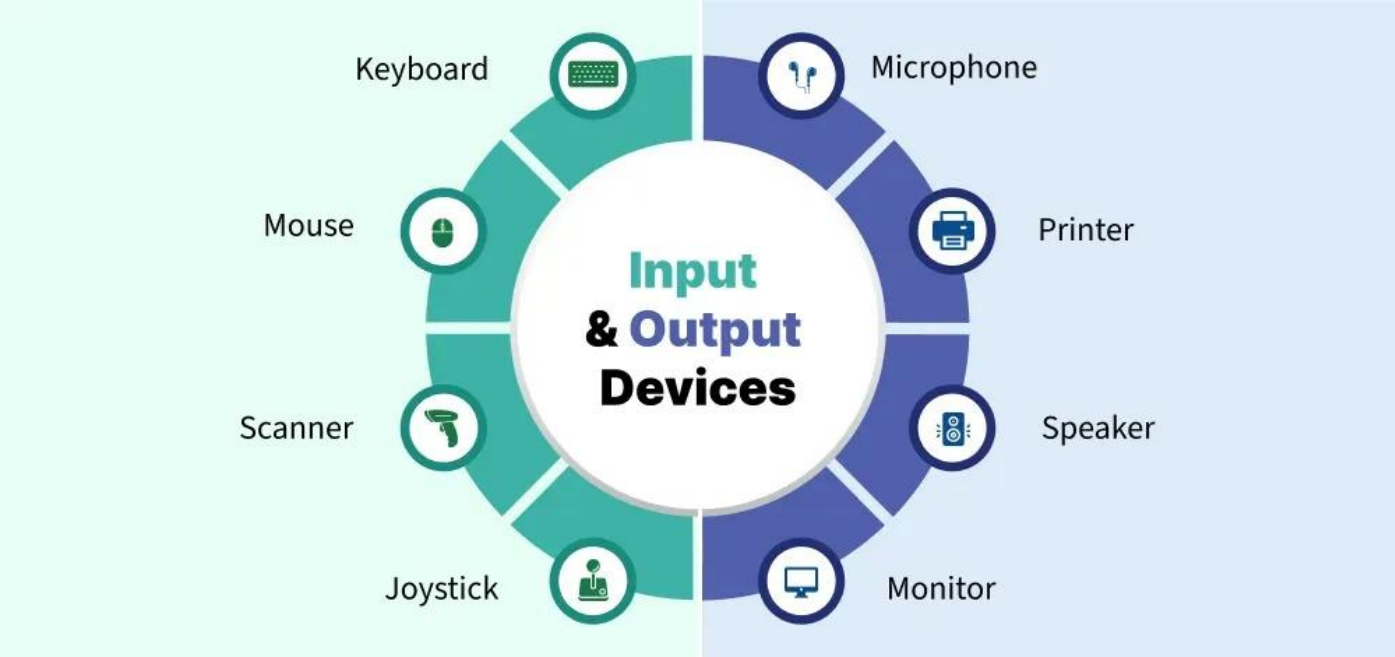
Disadvantages:

- More expensive than CDs and requires a compatible DVD drive.

Comparison of Major Storage Devices

Device	Type	Storage Capacity	Rewritable	Portability	Typical Use
Floppy Disk	Magnetic	1.44 MB	Yes	High	Small file transfers
Hard Disk	Magnetic	1 GB – several TB	Yes	Low	System and data storage
CD	Optical	650–700 MB	No (CD-RW – Yes)	High	Software, music, text
DVD	Optical	4.7–8.5 GB	No (DVD-RW – Yes)	High	Movies, multimedia, data backup

Unit : 3 Computer Architecture-Organization of Computer



17. Input and Output Devices

An **Input/Output (I/O) device** is any hardware that allows a user or another system to communicate with a computer.

- **Input Devices:** Send data or signals *to* the computer.
- **Output Devices:** Display or send processed data *from* the computer.
  Together, these devices create a bridge between the **user** and the **computer system**.

17.1 Input Devices

Input devices transfer data, commands, or signals to the CPU for processing. They can be classified into several types.

Classification of Input Devices

Category	Examples
Keyboard Devices	Standard Keyboard
Pointing Devices	Mouse, Joystick, Trackball, Light Pen
Composite Devices	Touch Screen, Scanner
Game Controllers	Joysticks, Game Pads
Visual Devices	Web Camera, Digitizer
Audio Input Devices	Microphone

17.1.1 Keyboard

- The **most common text-based input device** used to type data and commands.
- Layout similar to a **typewriter**, but with special and functional keys.
- Standard keyboards contain **101 to 108 keys**.

Types of Keys on Keyboard

Key Type	Purpose / Examples
Numeric Keys	Enter numeric data; include numbers (0–9) and mathematical operators.
Typing Keys	Alphabet (A–Z) and number keys for general typing.
Control Keys	Move cursor or control operations (Ctrl, Alt, Esc, Home, End).
Special Keys	Shift, Enter, Caps Lock, Tab, Delete, Spacebar, etc.
Function Keys	F1–F12; perform specific software-based functions.

17.1.2 Mouse

- A **pointing device** used to control the movement of a cursor on screen.
- Operates by rolling on a flat surface; used for **clicking, dragging, and selecting**.

Common Mouse Actions

Action	Function
Left Click	Selects an item.
Double Click	Opens a file or program.
Right Click	Displays shortcut menu.
Drag and Drop	Moves or copies selected items.

17.1.3 Joystick

- A **pointing device** with a stick that moves in multiple directions.
- Used for controlling motion on a screen (mainly in **gaming** and **CAD applications**).

17.1.4 Trackball

- Functions like an inverted mouse; user moves a **ball** with fingers to control the cursor.
- Common in **laptops** and **portable computers**.

17.1.5 Light Pen

- A **pen-shaped pointing device** used to draw or select directly on the screen.
- Contains a **photocell sensor** that detects light from the monitor.
- Commonly used in **graphic design** and **CAD** software.

17.1.6 Scanner

- Works like a **photocopier**, converting physical documents or images into digital form.
- Used for **digitizing text, photos, and graphics**.
- Enables **error-free data entry** and easy digital storage.

17.1.7 Optical Mark Reader (OMR)

- Detects **marks made by pencil or pen** on answer sheets or forms.
- Commonly used in **examination processing** and **surveys**.

17.1.8 Optical Character Reader (OCR)

- Converts **printed text into digital characters** using optical scanning.

- Saves scanned text directly into editable format.
- Used in **digitizing printed documents** like books or invoices.

17.1.9 Magnetic Ink Character Reader (MICR)

- Used in **banks** to read information on cheques printed with **magnetic ink**.
- Identifies cheque numbers and codes quickly and accurately.
- Provides **high-speed and error-free processing**.

17.1.10 Bar Code Reader

- Reads **data encoded in barcodes** (series of light and dark lines).
- Converts barcode patterns into **alphanumeric data**.
- Used in **supermarkets, libraries, and warehouses** for inventory and billing.

17.1.11 Web Camera (Webcam)

- Captures **video or images** in real time and transmits them to the computer.
- Used for **video calls, streaming, and conferencing**.
- Can be built-in (in laptops) or **connected via USB**.

17.1.12 Digitizer

- Converts **analog input (drawings, sketches)** into **digital data**.
- Example: **Graphics tablet** used by designers and architects.

17.1.13 Microphone

- Captures **sound signals** and converts them into digital form for processing.
- Used in **voice recording, video calls, and music applications**.

18. Output Devices

Output devices display or present the **processed results** from a computer system.
Output can appear as **text, images, graphics, sound, or video**.
They help users interpret and use the results of computer operations.

18.1 Common Output Devices

Output Device	Description / Function	Key Features / Uses
Monitor (VDU)	Displays images and text on screen using tiny dots called pixels.	Two main types: • **CRT Monitor:** Bulky, high power use, older type. • **Flat-Panel Display:** Lightweight, low power, used in laptops and modern systems.
Television (TV)	Displays video and audio signals from the computer or media source.	Modern TVs use **LED or Plasma displays** instead of CRTs; supports **multimedia presentations**.
Printer	Produces permanent output on paper (hard copy).	Divided into two main types: **Impact** and **Non-Impact** printers.

Impact Printers	Use mechanical striking through a ribbon to print on paper.	• **Low cost**, **noisy**, suitable for bulk printing. Examples: *Dot Matrix, Daisy Wheel, Line Printer (Drum/Chain)*.
Non-Impact Printers	Print without physical contact using heat, laser, or ink spray.	• **Quiet, fast, high-quality output**. Examples: *Laser Printer, Inkjet Printer, Thermal Printer*.
Laser Printer	Uses laser light to form characters and images with toner.	• High speed, excellent print quality, used in offices.
Inkjet Printer	Uses **ink spray technology** to produce text and color images.	• Suitable for **photo and color printing**, good quality but slower.
Plotter	Produces **graphics, charts, and designs** on paper using pens.	• Used in **engineering, CAD, and architectural drawings**.
Speakers	Convert **digital audio signals** into sound waves.	• Available as **wired or wireless (Bluetooth)**; used in media playback.
Headphones / Headsets	Small personal audio output device worn over ears.	• Used for **private listening**; often includes **microphones**.
Projector	Projects visual output from a computer onto a **large screen or surface**.	• Common in **classrooms, meetings, theatres**, and **presentations**.
Braille Reader	Specialized device for **visually impaired users**.	• Converts text on the screen into **tactile Braille characters** for reading by touch.
Video Card (Graphics Card)	A hardware component that enhances display quality and enables multiple screens.	• Installed on motherboard; improves **graphic and video performance**.
GPS (Global Positioning System)	Provides **location and navigation data** using satellites.	• Used in **vehicles, smartphones**, and **mapping applications** for accurate tracking.

18.2 Summary of Printer Types

Printer Type	Printing Method	Examples	Features
Impact Printers	Strike ribbon to print on paper.	Dot Matrix, Daisy Wheel, Line Printer	Low cost, noisy, used for bulk printing.
Non-Impact Printers	No physical contact with paper; use laser, ink, or heat.	Laser Printer, Inkjet Printer, Thermal Printer	Fast, quiet, high quality, supports multiple fonts.

18.3 Key Notes

- **Resolution:** Determines print or display sharpness; measured in **dots per inch (DPI)**.
- **Speed:** Measured as **CPS (characters per second)**, **LPM (lines per minute)**, or **PPM (pages per minute)**.
- **Modern Output Trend:** Devices are now **compact, energy-efficient**, and support **multimedia output** (audio, video, graphics).

19. Input-Output Devices (Hybrid Devices)

Some devices serve both as **input and output tools**, allowing data to flow **in both directions** between the

user and the computer.
They enhance efficiency by combining functionality—reducing the need for separate devices.

Key Functions of Hybrid Devices:

- Enable **data storage** and retrieval.
- Facilitate **communication** between systems.
- Support **interactive user interfaces**.

19.1 Common Examples of Hybrid Devices

Device	Acts As Input When…	Acts As Output When…	Common Uses / Features
Touchscreen	User touches or selects options on the display (input).	Displays images, menus, and visual content (output).	• Used in **Smartphones, Tablets, ATMs, Kiosks**. • Supports **multi-touch** (resistive and capacitive).
External Hard Drive	Computer reads or retrieves data stored on the drive.	User saves or writes files onto the drive.	• Used for **data backup**, **file transfer**, and **extra storage**. • Portable and connects via **USB or Thunderbolt**.
USB Flash Drive	Provides stored files to a computer.	Stores or receives files from a computer.	• Compact and **easy to carry**. • Used for **file sharing**, **system booting**, and **emergency backups**.

19.2 Key Characteristics of Hybrid Devices

- Combine both **data input and output** capabilities.
- **Portable, efficient**, and **user-friendly**.
- Essential in **modern computing** and **data transfer applications**.
- Support **bidirectional communication** between devices and systems.

Unit: 4 Software

20. Operating System (OS)

An **Operating System (OS)** is **system software** that manages **computer hardware, software resources, and provides services** for running applications.
It acts as a **bridge between the user and hardware**, coordinating input, output, memory, and processing.

20.1 Definition and Purpose

An **Operating System** is often defined as:

"The layer of software that manages a computer's resources for its users and their applications."

Main Purposes of an OS:

1. **Resource Management:**
    - Allocates CPU time, memory, and I/O devices among multiple programs.
    - Prevents one process from monopolizing system resources.
2. **Hardware Abstraction:**
    - Provides a simple interface for accessing complex hardware.
    - Uses **virtual memory** to simulate large memory space.
3. **Common Services:**
    - Offers file management, networking, device control, and system security.
    - Allows applications to run on different hardware without modification.

20.2 Functions of an Operating System

Function	Description
Process Management	Handles creation, scheduling, and termination of processes.
Memory Management	Allocates and deallocates memory; uses virtual memory.
File Management	Organizes data into files and directories; handles read/write operations.
Device Management	Manages device drivers and I/O coordination.
Security & Protection	Ensures system integrity through access control and isolation.
User Interface	Provides CLI (Command Line Interface) or GUI (Graphical User Interface).

20.3 Importance of OS in Modern Devices

Operating systems are present in nearly all devices that contain a processor, including:

- **Mobile devices:** Android, iOS, iPadOS
- **Desktops/laptops:** Windows, macOS, Linux
- **Servers:** Linux, UNIX
- **Embedded systems:** Routers, smart TVs, automotive systems

20.4 Global Market Share (as of 2024)

Operating System	Market Share	Category
Android	46%	Mobile OS
Windows	26%	Desktop OS
iOS / iPadOS	18%	Mobile OS
macOS	5%	Desktop OS
Linux	1%	Desktop OS, Server OS

*Linux distributions dominate in **servers** and **supercomputers**, while Android leads in **mobile systems**.*

20.5 Types of Operating Systems

Type	Description / Features	Examples / Use
Multicomputer OS	Manages multiple CPUs, each with its own memory; used in **cloud computing**.	Used in clusters; supports remote procedure calls and distributed memory.
Distributed OS	Coordinates multiple independent systems connected via a network.	Used in **networked and cloud environments** for shared computing.
Embedded OS	Designed for **specific embedded systems** (small, task-oriented).	Embedded Linux, QNX, VxWorks, RIOT, TinyOS.
Real-Time OS (RTOS)	Executes tasks within a **guaranteed time frame**.	**Hard RTOS:** Avionics, military. **Soft RTOS:** Multimedia, smartphones.
Hypervisor (Virtual Machine Monitor)	Enables multiple OSs to run on the same hardware.	VMware, Hyper-V, KVM.
Library OS (LibOS)	Converts OS services into **application-linked libraries** for lightweight systems.	Used in **unikernels** and **cloud environments** for high performance.

20.6 Components of an Operating System

Component	Description
Kernel	The core part that manages hardware and system resources.
System Programs	Utilities that assist in managing hardware and software.
Application Programs	End-user software (e.g., browsers, editors).
Device Drivers	Specialized programs that control I/O devices.

20.7 Installation and Operation

Operating systems can be:

- **Pre-installed (OEM)** on purchased computers.
- **Installed manually** via DVD or USB.
- **Run live** from external media (Live CD / Live USB).

20.8 Specialized OS Categories

- **Embedded OS:** Runs on limited hardware; no user-installed apps.
- **Real-Time OS:** Ensures precise event timing.

- **Security OS:** Focused on privacy, encryption, and sandboxing.
- **Lightweight OS:** Designed for low system resource usage.

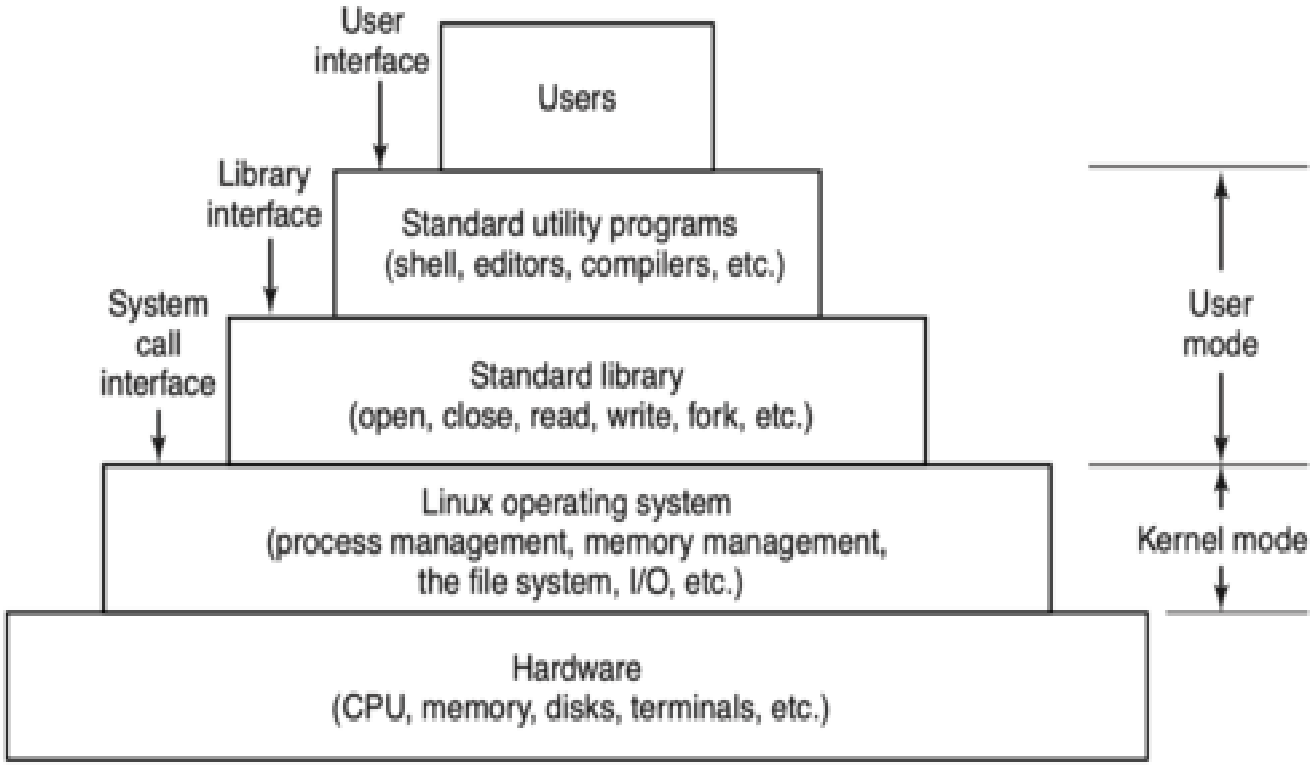

User interface
Users
Library interface
Standard utility programs (shell, editors, compilers, etc.)
System call interface
Standard library (open, close, read, write, fork, etc.)
Linux operating system (process management, memory management, the file system, I/O, etc.)
Hardware (CPU, memory, disks, terminals, etc.)
User mode
Kernel mode

Fig. OS Structure

List of Operating Systems

Name	Creator	Details
AIX	IBM	Initial release: 1986; Predecessor: UNIX System V Release 3; Version: 7.3; Release date: June 20, 2022; Cost: Commercial; License: Proprietary
AIX/370	IBM	Released: 1989; Predecessor: IX/370; Version: 2.1; Date: Feb 22, 1991; Cost: Non-free; License: Proprietary
AIX/ESA	IBM	Released: 1992; Predecessor: AIX/370, OSF/1; Version: 2.2; Date: Feb 26, 1993; Cost: Non-free; License: Proprietary
AmigaOS classic	Commodore, Haage & Partner, Hyperion	Released: 1985; Predecessor: TRIPOS; Version: 3.2.2.1; Date: Apr 23, 2023; Bundled with hardware; License: Proprietary/AROS Public License
AmigaOS 4	Hyperion Entertainment	Released: 2004; Predecessor: AmigaOS classic; Version: 4.1 Final Edition Update 2.1; Date: Jan 12, 2021; Cost: €29; License: Proprietary
Android	Android Inc., Google	Released: 2008; Version: 16; Date: June 10, 2025; Free; License: Apache-2.0 / GPL-2.0
ArcaOS	Arca Noae LLC	Released: 2017; Predecessor: OS/2; Version: 5.1.0; Date: Aug 27, 2023; Cost: $139–$249; License: Proprietary
Classic Mac OS	Apple Computer	Released: 1984; Predecessor: None; Version: 9.2.2; Date: 2000; Discontinued; License: Proprietary
ChromeOS	Google	Released: 2011; Predecessor: None; Version: 129.0.6668.99; Date: Oct 1, 2024; Bundled with hardware; License: Proprietary

ChromiumOS	Google	Released: 2009; Rolling release; Date: Oct 7, 2021; Free; License: BSD / GPL-2.0
DragonFly BSD	Matthew Dillon	Released: 2003; Predecessor: FreeBSD; Version: 6.4.0; Date: Jun 9, 2022; Free; License: BSD
eComStation	Serenity Systems, Mensys BV	Released: 2001; Predecessor: OS/2; Version: 2.1; Date: May 20, 2011; Discontinued; License: Proprietary
EPOC32	Psion PLC	Released: 1996; Version: ER5; Date: 1999; Discontinued; License: Proprietary
EulerOS	Huawei	Released: 2021; Version: V2.0SP8; Date: 2022; Free; License: Open source
FreeBSD	The FreeBSD Project	Released: 1993; Predecessor: 386BSD; Version: 14.1; Date: Jun 4, 2024; Free; License: BSD
Genode	Genode Labs	Released: 2008; Version: 25.08; Date: Aug 28, 2025; Free; License: AGPL-3.0
GhostBSD	Eric Turgeon	Released: 2009; Predecessor: FreeBSD; Version: 21.05; Date: May 13, 2021; Free; License: BSD
Haiku	Haiku Inc.	Released: 2002; Predecessor: BeOS R5; Version: R1/Beta 5; Date: Sep 13, 2024; Free; License: MIT
HP-UX	Hewlett Packard Enterprise	Released: 1983; Predecessor: UNIX System V; Version: 11i v3 2023; Date: May 2022; Cost: $400; License: Proprietary
HarmonyOS	Huawei	Released: 2019; Predecessor: OpenHarmony, LiteOS; Version: 5.0.0.102; Date: Oct 23, 2024; Bundled; License: Proprietary / open components
IBM i	IBM	Released: 1988; Predecessor: CPF, SSP; Version: 7.5; Date: May 10, 2022; Bundled; License: Proprietary
Inferno	Bell Labs	Released: 1996; Predecessor: Plan 9; Version: Fourth Edition; Date: Jun 30, 2009; Free; License: MIT
iOS	Apple Inc.	Released: 2007; Predecessor: macOS; Version: 26.0.1; Date: Sep 29, 2025; Bundled; License: Proprietary / open-source core
IRIX	SGI	Released: 1988; Predecessor: UNIX System V; Version: 6.5.30; Date: 2006; Discontinued; License: Proprietary
Linux	Linus Torvalds & community	Released: 1991; Predecessor: None; Version: 6.17.5 (kernel); Date: Oct 23, 2025; Free; License: GPL-2.0
macOS	Apple Inc.	Released: 2001; Predecessor: NeXTSTEP, BSD; Version: 26.0.1; Date: Sep 29, 2025; Bundled; License: Proprietary / open-source core
NetBSD	The NetBSD Project	Released: 1993; Predecessor: 386BSD; Version: 9.3; Date: Aug 4, 2022; Free; License: BSD
OpenBSD	OpenBSD Project	Released: 1996; Predecessor: NetBSD 1.0; Version: 7.7; Date: Apr 28, 2025; Free; License: ISC
ReactOS	ReactOS Team	Released: 1998; Predecessor: Windows NT (clone); Version: 0.4.14; Date: Apr 9, 2020; Free; License: GPL-2.0
RISC OS 5	Castle Technology, RISC OS Open	Released: 2002; Predecessor: RISC OS 4; Version: 5.28; Date: 2020; Free; License: Apache-2.0
Solaris	Sun Microsystems (Oracle)	Released: 1992; Predecessor: SunOS; Version: 11.4; Date: Aug 28, 2018; Free for development; License: CDDL
Windows (NT family)	Microsoft	Released: 1993; Predecessor: OS/2 & Windows 3.1x; Version: Windows 11 (24H2); Date: Jun 15, 2024; License fee; License: Proprietary / Source-available

z/OS	IBM	Released: 2000; Predecessor: OS/390; Version: 3.2; Date: Sep 30, 2025; Cost based on processor; License: Proprietary

List of OS Released by Microsoft

Name	Release date	Details
Windows 1.0	1985-11-20	Latest: 1.04 (1987-04-08); Support: Unsupported (2001-12-31); Codename: Interface Manager; Requires: DOS 2.0+; Arch: 16-bit; Target: Desktops.
Windows 2.0	1987-09-08	Latest: 2.03 (1987-12-09); Support: Unsupported (2001-12-31); Requires: DOS 3.0+; Arch: 16-bit; Target: Desktops.
Windows 2.1x	1988-05-27	Latest: 2.11 (1989-03-13); Support: Unsupported (2001-12-31); Requires: DOS 3.0+; Arch: 16-bit; Target: Desktops.
Windows 3.0	1990-05-22	Latest: 3.0a with Multimedia Extensions (1991-10-20); Support: Unsupported (2001-12-31); Requires: DOS 3.1+; Arch: 16-bit; Editions: Windows 3.0 / 3.0a (MME); Target: Desktops.
Windows 3.1x	1992-04-06	Latest: 3.11 (1993-12-31); Support: Unsupported (2001-12-31); Codenames: Janus / Kato / Snowball / Sparta; Requires: DOS 3.3+; Arch: 16-bit* (partial 32-bit via Win32s); Editions: Windows 3.1, WfW 3.1/3.11, Windows 3.2 (China); Target: Desktops.
Windows 95	1995-07-14	RTM build: 950; Latest: 4.00.950C OSR2.5 (1997-11-26); Support: Unsupported (2001-12-31); Codename: Chicago; MS-DOS: 7.0–7.1; Kernel: Monolithic / Hybrid 16/32-bit; Editions: Retail/OSR variants; Target: Desktops.
Windows 98	1998-05-15	RTM: 1998; Latest: 4.10.1998; Support: Unsupported (2006-07-11); Codename: Memphis; MS-DOS: 7.1; Kernel: Monolithic / Hybrid 16/32-bit; Target: Desktops.
Windows 98 SE	1999-05-05	RTM: 2222; Latest: 4.10.2222A (2000-02-25); Support: Unsupported (2006-07-11); MS-DOS: 7.1; Kernel: Hybrid 16/32-bit; Target: Desktops.
Windows Me	2000-06-19	RTM: 3000; Latest: 4.90.3000 (2000-09-14); Support: Unsupported (2006-07-11); Codename: Millennium; MS-DOS: 8.0; Kernel: Monolithic / Hybrid 16/32-bit; Target: Desktops.
Windows NT 3.1	1993-07-27	Version: 3.1; RTM build: 528; Latest: 528 (SP3) (1994-11-10); Support: Unsupported (2000-12-31); Codename: New Technology / OS/2; Arch: IA-32, DEC Alpha, MIPS; Editions: Workstation, Advanced Server.
Windows NT 3.5	1994-09-21	Version: 3.5; RTM build: 807; Latest: 807 (SP3) (1995-06-21); Support: Unsupported (2001-12-31); Codename: Daytona; Arch: IA-32, DEC Alpha, MIPS, PowerPC; Editions: Workstation, Server.

Windows NT 3.51	1995-05-30	Version: 3.51; RTM build: 1057; Latest: 1057 (SP5) (1996-09-19); Support: Unsupported (2001-12-31); Arch: IA-32, DEC Alpha, MIPS, PowerPC; Editions: Workstation, Server.
Windows NT 4.0	1996-07-31	Version: 4.0; RTM build: 1381; Latest: 1381 (SP6a) (1999-11-30); Support: Unsupported (2004-06-30); Extended updates issued until 2006; Codenames: Cairo / Hydra / Impala; Arch: IA-32, DEC Alpha, MIPS, PowerPC; Editions: Workstation, Server, Terminal Server, Embedded.
Windows 2000	1999-12-15	Version: 5.0; RTM build: 2195; Latest: SP4 Rollup 1 v2 (2005-09-13); Support: Unsupported (2010-07-13); Codename: Windows NT 5.0 / Janus (64-bit); Arch: IA-32; Editions: Professional, Server, Advanced Server, Datacenter, Embedded.
Windows XP	2001-08-24	Version: 5.1; RTM build: 2600; Latest: 2600 (SP3) (2008-04-21); Support: Unsupported (2014-04-08); Codename: Whistler; Arch: IA-32, IA-64, x86-64; Editions: Home, Professional, Media Center, Tablet PC, Starter, Embedded.
Windows Server 2003	2003-04-24	Version: 5.2; RTM build: 3790; Latest: 3790 (SP2) (2007-03-13); Support: Unsupported (2015-07-14); Codenames: Whistler Server / Windows.NET Server; Arch: IA-32, IA-64, x86-64; Editions: Standard, Enterprise, Datacenter, Web, Storage, SBS, Compute Cluster.
Windows Server 2003 R2	2005	Version: 5.2; RTM build: 3790; Latest: (varied); Support: Unsupported (2015-07-14); Arch: IA-32, x86-64; Editions: Server (various) — targeted at server/HPC/embedded appliances.
Windows Fundamentals for Legacy PCs	2006-07-08	Based on XP (5.1 / 2600); Support: Unsupported (2014-04-08); Codenames: Eiger, Mönch; Arch: IA-32; Edition: Fundamentals for Legacy PCs; Target: Desktop (low-spec machines).
Windows Vista	2006-11-30	Version: 6.0; RTM build: 6000; Latest: 6002 (SP2) (2009-04-28); Support: Unsupported (2017-04-11); Codename: Longhorn; Arch: IA-32, x86-64; Editions: Starter, Home Basic/Premium, Business, Enterprise, Ultimate; Target: Desktop/Workstation.
Windows Home Server	2007-11-04	Based on 5.2 (Server); Support: Unsupported (2013-01-08); Codenames: Q, Quattro; Arch: IA-32, x86-64; Edition: Home Server; Target: Home server.
Windows Server 2008	2008-02-04	Version: 6.0; RTM build: 6001; Latest: 6002 (SP2) (2009-04-28); Support: Out of extended support (2020-01-14) with limited paid Premium Assurance until Jan 13, 2026; Codename: Longhorn Server; Arch: x86-64, IA-64, IA-32; Editions: WebCore, Standard, Enterprise, Datacenter, Small Business, HPC, Hyper-V, Foundation, Storage; Target: Server.
Windows 7	2009-07-22	Version: 6.1; RTM build: 7600; Latest: 7601 (SP1) (2011-02-22); Support: Unsupported (2020-01-14); Codename: 7; Arch: IA-32,

		x86-64; Editions: Starter, Home Basic/Premium, Professional, Ultimate, Enterprise; Target: Desktop/Workstation.
Windows Server 2008 R2	2009-07-22	Version: 6.1; RTM build: 7600; Latest: 7601 (SP1); Support: Out of extended support (2020-01-14), limited paid Premium Assurance through Jan 13, 2026; Codename: 7 Server; Arch: x86-64, IA-64; Editions: StandardCore, EnterpriseCore, DatacenterCore, WebCore.
Windows Home Server 2011	2011-04-06	Version: 6.1; RTM build: 8400; Support: Unsupported (2016-04-12); Codename: Vail; Arch: x86-64; Edition: Home Server.
Windows Server 2012	2012-08-01	Version: 6.2; RTM build: 9200; Support: Unsupported (2023-10-10); Codename: 8 Server; Arch: x86-64; Editions: Foundation, Essentials, Standard, Datacenter, Storage.
Windows 8	2012-08-01	Version: 6.2; RTM build: 9200; Support: Unsupported (2016-01-12); Codename: 8; Arch: IA-32, x86-64, ARMv7; Editions: Windows 8, Pro, Enterprise, Windows RT; Target: Desktop/Multi-touch.
Windows 8.1	2013-08-27	Version: 6.3; RTM build: 9600; Support: Unsupported (2023-01-10); Codename: Blue; Arch: IA-32, x86-64, ARMv7; Editions: Windows 8.1, Pro, Enterprise, Windows RT 8.1.
Windows Server 2012 R2	2013-10-18	Version: 6.3; RTM build: 9600; Support: Out of extended support (2023-10-10) with paid ESU available until Oct 13, 2026; Codename: Blue Server; Arch: x86-64; Editions: Foundation, Essentials, Standard, Datacenter, Storage.
Windows 10 — Original Release	2015-07-29	Version: 10.0; RTM build: 10240; Support: Unsupported (2017-05-09); Codename: Threshold; Arch: IA-32, x86-64, ARMv7; Editions: Home, Pro, Education, Enterprise.
Windows 10 — November Update (1511)	2015-11-12	Build: 10586; Support: Unsupported (2017-10-10); Codename: Threshold 2; Editions: Home, Pro, Education, Enterprise.
Windows 10 — Anniversary Update	2016-08-02	Build: 14393; Support: Unsupported (2018-04-10); Codename: Redstone; Editions: Home, Pro, Pro Education.
Windows 10 — Creators Update	2017-04-11	Build: 15063; Support: Unsupported (2018-10-09); Codename: Redstone 2; Editions: Home, Pro, Pro Education.
Windows 10 — Fall Creators Update	2017-10-17	Build: 16299; Support: Unsupported (2019-04-09); Codename: Redstone 3; Arch: IA-32, x86-64, ARM64; Editions: Home, Pro, Pro for Workstations, Pro Education.
Windows 10 — April 2018 Update	2018-04-30	Build: 17134; Support: Unsupported (2019-11-12); Codename: Redstone 4; Editions: Home, Pro, Pro for Workstations, Pro Education.

Windows 10 — October 2018 Update	2018-10-02	Build: 17763; Support: Unsupported (2020-11-10); Codename: Redstone 5; Editions: Home, Pro.
Windows 10 — May 2019 Update	2019-05-21	Build: 18362; Support: Unsupported (2020-12-08); Codename: 19H1; Editions: Home, Pro, Education, Enterprise.
Windows 10 — November 2019 Update	2019-11-12	Build: 18363; Support: Unsupported (2021-05-11); Codename: 19H2; Editions: Home, Pro, Education, Enterprise.
Windows 10 — May 2020 Update	2020-05-27	Build: 19041; Support: Unsupported (2021-12-14); Codename: 20H1; Editions: Home, Pro, Education, Enterprise.
Windows 10 — October 2020 Update	2020-10-20	Build: 19042; Support: Unsupported (2022-05-10); Codename: 20H2; Editions: Home, Pro, Education, Enterprise.
Windows 10 — May 2021 Update	2021-05-18	Build: 19043; Support: Unsupported (2022-12-13); Codename: 21H1; Editions: Home, Pro, Education, Enterprise.
Windows 10 — November 2021 Update	2021-10-16	Build: 19044; Support: Unsupported (2023-06-13); Codename: 21H2; Editions: Home, Pro, Education, Enterprise.
Windows 10 — 2022 Update (22H2)	2022-10-18	Build: 19045; Support: Unsupported (2025-10-14) for many SKUs; ESU was offered (paid) through Oct 13, 2026 (consumers) / Oct 10, 2028 (business/schools); Editions: Home, Pro, Pro for Workstations, Enterprise.
Windows Server 2016	2016-09-26	Version: 10.0; Build: 14393; Support: On extended support (2027-01-12); Codename: Redstone Server; Arch: x86-64; Editions: Essentials, Standard, Datacenter, MultiPoint, Storage.
Windows Server 2019	2018-10-02	Version: 10.0; Build: 17763; Support: On extended support (2029-01-09); Codename: Redstone 5 Server; Arch: x86-64; Editions: Essentials, Standard, Datacenter.
Windows Server 2022	2021-08-18	Version: 10.0; Build: 20348; Support: Mainstream until 2026-10-13; Extended until 2031-10-14; Codename: 21H2 Server; Arch: x86-64; Editions: Essentials, Standard, Datacenter, Datacenter for Azure.
Windows 11 — Original Release (21H2)	2021-10-05	Version: 10.0; Build: 22000; Support: Unsupported (2023-10-10) for some SKUs; Codename: Sun Valley; Arch: x86-64, ARM64; Editions: Home, Pro, Pro for Workstations, Pro Education, SE; Target: Desktop/Multi-touch.
Windows 11 — 2022 Update (22H2)	2022-09-20	Build: 22621; Support: Unsupported (2025-10-14) for some SKUs; Editions: Home, Pro, Pro for Workstations, Pro Education, SE; Arch: x86-64, ARM64.

Windows 11 — 2023 Update (23H2)	2023-10-31	Build: 22631; Support: Supported (until 2026-11-10 for some SKUs); Editions: Home, Pro, Pro for Workstations, Pro Education, SE.
Windows 11 — 2024 Update (24H2)	2024-10-01	Build: 26100; Support: Supported (mainstream until 2026-10-13 for many SKUs); Editions: Home, Pro, Pro for Workstations, Pro Education, SE.
Windows 11 — 2025 Update (25H2)	2025-09-30	Build: 26200; Support: Supported (varies by SKU; many SKUs supported until 2027-10-12); Editions: Home, Pro, Pro for Workstations, Pro Education, SE.
Windows Server 2025	2024-11-01	Version: 10.0; Build: 26100; Support: Mainstream until 2029-10-09; Extended until 2034-10-10; Codename: 24H2 Server; Arch: x86-64, ARM64; Editions: Essentials, Standard, Datacenter, Datacenter for Azure.

21. UNIX Operating System

Definition:
UNIX is a family of **multitasking, multi-user operating systems** developed in **1969** at **AT&T's Bell Labs** by **Ken Thompson, Dennis Ritchie**, and others.
It became the foundation for many later systems, including **Linux**, **BSD**, and **macOS**.

21.1 Key Features of UNIX

Feature	Description
Multitasking	Can run multiple programs simultaneously.
Multi-user	Supports multiple users accessing the system at once.
Portability	Written in **C language**, making it easy to modify and run on different hardware.
Security	Provides user authentication and file permission levels.
Hierarchical File System	Organizes data in a tree-like directory structure.
Device Independence	Treats devices as files, enabling uniform access.
Shell Interface	Provides command-line interface and shell scripting for automation.
Modularity	Follows the **"Unix Philosophy"** — small programs doing specific tasks that can be combined flexibly.

21.2 UNIX Philosophy

UNIX emphasizes **simplicity, modularity, and reusability**.
According to its philosophy:

"Each program should do one thing well and work together with others through simple interfaces."

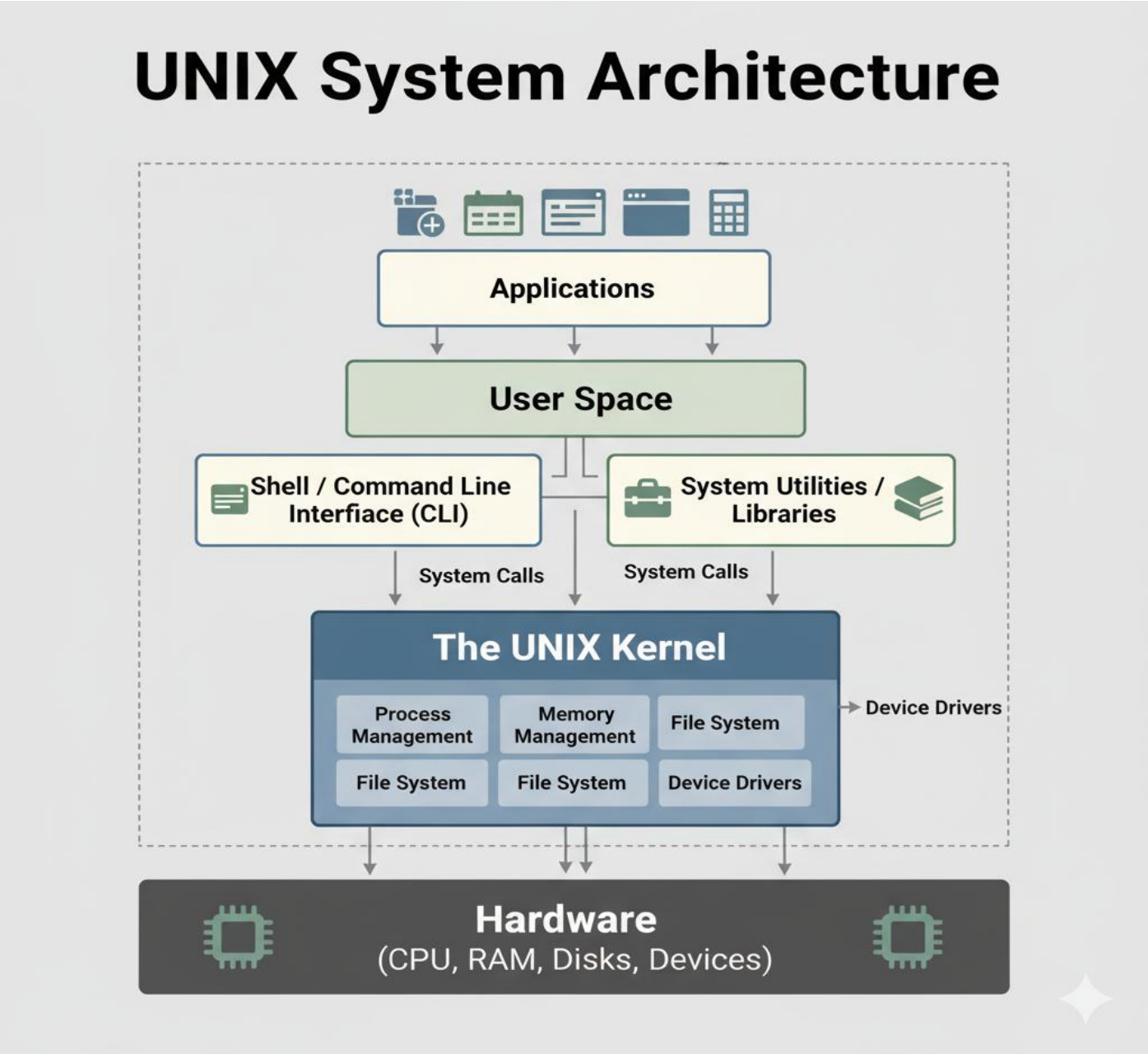
Key concepts include:

- **Plain text files** for data storage.
- **Pipes and filters** for inter-process communication.
- **Shell scripting** to combine tools into complex workflows.

21.3 History and Development

Year / Version	Milestone / Event
1969	UNIX developed at Bell Labs by Thompson and Ritchie.
1970–1973	Written in **C language**, becoming the first **portable OS**.
1975–1979	University and research use expands (Version 7 UNIX).
1982	AT&T releases **UNIX System III** commercially.
1983	**System V** released — widely adopted by major vendors.
1989	**System V Release 4 (SVR4)** merges BSD, SunOS, and Xenix features.
1990s	UNIX trademark sold to **The Open Group**; introduction of **FreeBSD**, **Linux**, and **macOS**.
2000s–Present	UNIX principles dominate in servers, mobile OS (Android/iOS), and macOS systems.

21.4 UNIX Architecture



Component	Function
Kernel	Core of UNIX; manages CPU, memory, and devices.
Shell	Command interpreter; user interface between user and kernel.
File System	Organizes and manages files in directories.
Utilities / Commands	Perform basic functions (e.g., cp, mv, ls, cat).
Applications	User programs and software running on the system.

Architecture Overview:
User → Shell → Kernel → Hardware

21.5 Major UNIX Variants

Variant	Developer / Organization	Base Type
BSD (Berkeley Software Distribution)	University of California, Berkeley	Research UNIX
System V (SVR4)	AT&T and Sun Microsystems	Commercial UNIX
Solaris	Sun Microsystems (Oracle)	System V-based
HP-UX	Hewlett-Packard	System V-based
AIX	IBM	System V-based
Xenix	Microsoft (early UNIX version)	System V-based
FreeBSD, OpenBSD, NetBSD	Open-source communities	BSD-based

21.6 Importance and Influence

- **C Programming Language:** UNIX and C evolved together — both portable and efficient.
- **Networking and Internet:** UNIX's client–server model influenced the development of the Internet.
- **Open Source Evolution:** Inspired modern UNIX-like systems such as **Linux**, **FreeBSD**, and **macOS**.
- **Server Dominance:** UNIX and UNIX-like systems power most **web servers**, **supercomputers**, and **cloud infrastructure**.

21.7 Modern Status

- **Current Use:** Widely used in servers, mainframes, and high-end workstations.
- **UNIX Certification:** Managed by **The Open Group** under the **Single UNIX Specification (SUS)**.
- **Market Leaders:**
    - **IBM AIX**, **Oracle Solaris**, and **HP-UX** remain major commercial UNIX systems.
    - **Linux** and **macOS** are popular **UNIX-like** successors.

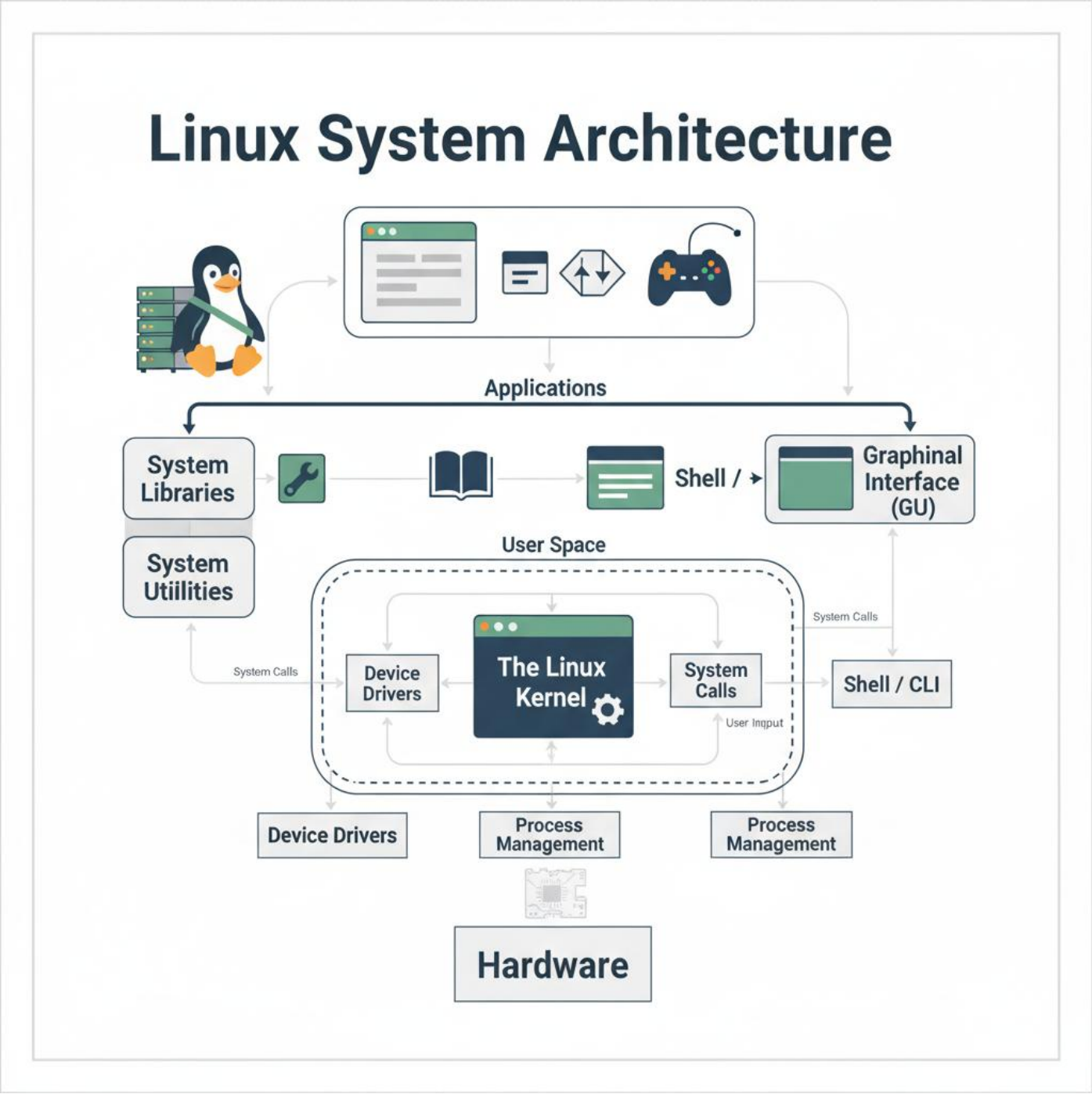
Unix Components

Component	Typical location / files	Details & examples
Kernel (V7 canonical structure)	/usr/sys (source tree)	Subcomponents: **conf** (configuration & boot code), **dev** (device drivers), **sys** (kernel: memory, scheduling, syscalls), **h** (header files). Examples: conf, dev, sys, h.
Development environment	/usr/src, /lib, /usr/lib, /usr/include	Tools to rebuild system from source: **ed** (text editor), **cc** (C compiler), **as** (assembler), **ld** (linker), **make**

		(build manager), **libc** (system library), **include** (headers). Examples: ed, cc, make, libc.
Libraries	/lib, /usr/lib	Runtime and utility libraries (libc/stdio, libm, others). V7 introduced stdio. Examples: libc, libm, stdio.
Other languages	/usr/bin, /usr/local/bin (compilers/tools)	Additional language toolchains: **Fortran-77**, **bc/dc** (calculator), **awk** (text processing); BSD added Pascal; modern systems often include **GCC**. Examples: Fortran-77, awk, GCC, bc.
Other tools / toolchain utilities	/usr/bin, /usr/lib	Compiler-support and binary tools: **ar** (archive), **nm** (symbol lister), **lex**, **yacc**, debuggers (e.g., **gdb**). Examples: ar, nm, lex, yacc.
Shell (command interpreter)	/bin/sh, /bin/*	Primary programmable interface: **sh** historically; later shells provide features & scripting. Examples (common shells): sh, bash, ksh, csh.
Utilities — core toolkit	/bin, /usr/bin, /sbin	Small modular programs for everyday tasks. — **System utilities** (mkfs, fsck, mount, init). — **User utilities** (passwd, kill, who, chown). Examples: cp, ls, grep, find; system examples: mkfs, fsck, init.
Document formatting / typesetting	/usr/bin, /usr/lib/troff*	Early Unix included nroff/troff toolchain and preprocessors: **nroff**, **troff**, **tbl**, **eqn**, **refer**, **pic**; later additions: **TeX**, **Ghostscript**. Examples: troff, tbl, eqn, TeX.
Graphics / plotting	device-specific paths, /usr/bin	Plot subsystem for device-independent vector plots + device interpreters; modern systems add **X11** and **OpenGL** support. Examples: plot subsystem, X11, OpenGL.
Communications	/usr/bin, /usr/lib/uucp, /etc	Inter-user and inter-system comms: early mail/write; **UUCP** for store-and-forward; later BSD added **TCP/IP** utilities (telnet, ftp, rsh, ping). Examples: mail, write, uucp, telnet/ftp.
Documentation	man trees (/usr/man, /usr/share/man), /usr/doc	Machine-readable documentation included with system: **man pages** (commands, syscalls, headers) and longer docs/typeset volumes (C language, troff manuals). Examples: man pages, system design docs, troff-formatted manuals.

22. Linux Operating System

Definition:
Linux is an **open-source, Unix-like operating system** based on the **Linux kernel**, first released by **Linus Torvalds** on **September 17, 1991**.
It follows the principles of UNIX but is **free, portable, and community-driven**, distributed under the **GNU General Public License (GPL)**.



Linux Structure

22.1 Key Features of Linux

Feature	Description
Open Source	Freely available to use, modify, and distribute.
Unix-like Architecture	Based on the UNIX philosophy of modular, command-based design.
Multitasking & Multi-user	Supports multiple programs and users simultaneously.
Security	Strong permissions, encryption, and authentication system.
Portability	Runs on a wide variety of hardware platforms — from PCs to supercomputers.
Networking Support	Designed for networked environments; ideal for servers and clusters.
Stability & Performance	Highly reliable; minimal crashes even under heavy loads.

Customizability	Can be tailored to specific needs — from minimal systems to full-featured desktops.

22.2 History and Origin

Year / Event	Description
1983	**GNU Project** started by **Richard Stallman** to create a free Unix-like OS.
1987	**Minix**, a teaching OS by **Andrew Tanenbaum**, influenced Linux development.
1991	**Linus Torvalds** developed and released the first version of the **Linux kernel**.
1992	Linux was re-licensed under the **GNU GPL**, making it fully open source.
Mid-1990s	Rapid adoption by universities, programmers, and later commercial enterprises.
2000s–Present	Became the **dominant OS** for servers, supercomputers, and Android devices.

22.3 Components of a Linux Distribution

Component	Function
Linux Kernel	The core managing CPU, memory, and devices.
GNU Tools	Provide compilers, libraries, and system utilities.
Shell	Command-line interface (e.g., Bash, Zsh).
Bootloader	Starts the system (e.g., GRUB, LILO).
Package Manager	Handles software installation (e.g., APT, YUM, DNF).
Display Server	Manages graphical display (e.g., X11, Wayland).
Desktop Environment	Provides GUI (e.g., GNOME, KDE Plasma, XFCE).

22.4 Major Linux Distributions

Category	Examples	Description
Community-Based	Ubuntu, Fedora, Debian, Arch Linux, Linux Mint	Freely available, community-supported, frequent updates.
Commercial / Enterprise	Red Hat Enterprise Linux (RHEL), SUSE Linux Enterprise (SLES), Oracle Linux	Supported by companies for servers and businesses.
Specialized / Lightweight	Puppy Linux, Kali Linux, ChromeOS	Designed for specific uses like penetration testing or education.

22.5 Linux Architecture

Layer	Description
Hardware	The physical devices (CPU, memory, disk).
Kernel	Core component managing system resources.
System Libraries	Allow communication between kernel and applications.
System Utilities	Perform basic operations (file, device, and process management).
Shell / User Interface	Interprets user commands.
Applications	End-user programs and tools.

Structure:

User → Shell → Kernel → Hardware

22.6 Linux vs UNIX

Basis	Linux	UNIX
Source Code	Open source and free under GPL.	Proprietary (mostly commercial).
Cost	Free to use and modify.	Paid or licensed.
Platform Support	Runs on almost any hardware.	Limited to specific hardware.
User Interface	GUI + Command Line.	Mostly Command Line.
Usage	Servers, desktops, mobile (Android).	Mainframes, servers, workstations.
Security	High (community-reviewed).	High but less customizable.
Development	Community-driven.	Vendor-driven.

22.7 Advantages of Linux

- **Free and Open Source:** No licensing cost.
- **Stable and Secure:** Rarely crashes; protected file system.
- **Highly Customizable:** Users can modify kernel and components.
- **Wide Hardware Support:** Runs on old and new systems.
- **Multiplatform Availability:** Supports desktops, servers, mobile, and embedded systems.
- **Strong Developer Community:** Continuous updates and patches.

22.8 Applications of Linux

- **Servers:** Web, database, and mail servers.
- **Networking:** Routers, firewalls, and cloud infrastructure.
- **Desktop Systems:** Ubuntu, Fedora for home and office use.
- **Mobile Devices:** Android (based on Linux kernel).
- **Embedded Systems:** Smart TVs, IoT devices, vehicles, satellites, etc.
- **Supercomputers:** Used in **100%** of the world's top 500 supercomputers

23. Programming Languages

Definition:
A **programming language** is a formal set of rules and syntax used to write instructions that a computer can execute.
It acts as a **bridge between human logic and machine operations**, enabling developers to build **software, websites, and systems**.

Purpose:

- To communicate with computers.
- To create problem-solving algorithms.
- To control hardware and manage data flow efficiently.

23.1 Key Concepts

Term	Description	Example / Note
Syntax	The grammar or structure rules of a programming language.	C uses {} for blocks; Python uses indentation.
Data Types	Defines the type of data (numeric, text, logical, etc.).	int, float, string, boolean.
Variables	Named storage that holds data values.	score = 100
Operators	Symbols that perform operations on data.	+, -, *, /, ==, >
Control Statements	Control program flow or decisions.	if-else, for, while, switch
Functions	Blocks of code designed to perform a specific task.	def area():, void main()
Libraries / Frameworks	Pre-written code collections for reuse.	Python's **pandas**, JavaScript's **React**
Paradigms	Different programming styles or approaches.	**Procedural (C)**, **Object-Oriented (Java)**, **Functional (Haskell)**

23.2 Types of Programming Languages

Type	Description	Examples
High-Level Languages	Close to human language; easy to learn and portable. Require a compiler or interpreter.	Python, Java, JavaScript, C#, Swift.
Assembly Languages	Symbolic instructions that represent low-level machine operations. Require an **assembler**.	Assembly (ASM).
Low-Level Languages	Close to binary (machine code), directly interact with hardware. Fast but hard to write.	Machine Code (0s and 1s).

23.3 Programming Paradigms

Paradigm	Description	Languages / Example
Procedural	Focuses on a sequence of steps or procedures.	C, Pascal
Object-Oriented (OOP)	Organizes code into objects (data + behavior).	Java, Python, C++
Functional	Treats computation as mathematical functions; avoids changing states.	Haskell, Lisp
Scripting	Used for automating tasks or web functions.	Python, JavaScript
Logic-Based	Uses formal logic to derive conclusions from facts.	Prolog

23.4 Popular Programming Languages and Their Uses

Language	Key Features	Common Uses
Python	Simple, versatile, huge library support.	Web apps, data science, AI, automation.
Java	Platform-independent, secure, object-oriented.	Enterprise apps, Android apps, backend systems.

JavaScript	Runs in browsers; adds interactivity to websites.	Web development, UI design, front-end scripting.
C++	Fast, powerful, object-oriented with low-level control.	Game engines, operating systems, simulations.
C#	Microsoft's language for Windows apps and games.	Desktop apps, Unity game development.
Swift	Modern, safe, and efficient.	iOS and macOS application development.
Go (Golang)	Developed by Google; high-performance and scalable.	Cloud services, networking, backend servers.
Ruby	Clean syntax, web-oriented.	Web apps using **Ruby on Rails** framework.

23.5 Translation of Programs

Translator Type	Function	Example
Compiler	Converts the entire code into machine language before execution.	C, C++
Interpreter	Translates and executes code line by line.	Python, JavaScript
Assembler	Converts assembly code into machine code.	Assembly language

.

24. Flowchart

Definition:
A **Flowchart** is a **diagrammatic representation of an algorithm**, using specific symbols to depict the **flow of logic, operations, and decisions** in a process or program.

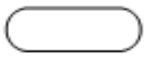
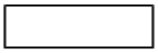
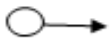
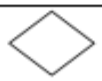
It helps visualize **how data flows**, **what operations occur**, and **in what sequence** — making the logic easier to understand and implement in any programming language.

24.1 Characteristics of a Flowchart

- **Graphical Representation:** Uses symbols and arrows to show steps in order.
- **Language Independent:** Same flowchart can be used for coding in any language.
- **Logical Visualization:** Shows what actions take place and in what sequence.
- **Stepwise Design:** Breaks a complex process into smaller, understandable parts.
- **Basis for Coding:** Acts as a blueprint before writing an actual program.

24.2 Purpose of Flowchart

- To **analyze and document** a process.
- To **illustrate logical flow** of algorithms clearly.
- To **simplify debugging** and testing by visualizing program structure.
- To **communicate program logic** among programmers and users effectively.

Flowcharting Symbols		
Symbol	**Name**	**Description**
	Terminal	Defines the starting and ending point of a flowchart
	Initialization	Initialization of memory space for data processing.
	Input/output	Input of data for processing and printing of processed data.
	Process	Manipulation of data (assignments and mathematical computations)
	Flow lines	Defines logical sequence of the program.
	On-page connector	Creates a cross-reference from one process to another on the same page of the flow chart
	Off-page connector	Creates a cross-reference from a process on one page to a process on another page.
	Decision	Processes conditions using relational operators. Used for trapping and filtering data.

24.3 Advantages of Flowchart

- Provides **clear and visual understanding** of logic.
- Simplifies **program debugging and modification**.
- Helps in **efficient coding** and **program maintenance**.
- Useful for **training new programmers**.

24.4 Disadvantages of Flowchart

- **Time-consuming** to design for large or complex programs.
- **Difficult to update** once drawn.
- Requires **skill and precision** in symbol arrangement.

24.5 Common Flowchart Symbols and Their Meanings

Symbol	Name / Function	Description / Purpose
■	**Terminator (Start / Stop)**	Indicates the **beginning or end** of a program or process. Usually labeled *START* or *END*.
□	**Process / Instruction Box**	Represents a **process or action**, such as calculations or assignments. Example: Total = Price × Quantity.
△	**Decision (Diamond)**	Represents a **decision or condition** with two branches — *Yes/No* or *True/False*.
⬒	**Input / Output (Parallelogram)**	Denotes **input** (data entry) or **output** (display of results). Example: Read A, B or Print Sum.

	Flow Lines / Arrows	Show the **direction of program control flow** from one step to another.
	Connector (Circle)	Used to **connect flow lines** when space is limited or flow continues on another page.
	Predefined Process (Subroutine)	Represents a **subroutine or function** used elsewhere in the program.
	Annotation / Comment	Used for **explanatory notes** or remarks within a flowchart.

(Note: shapes are symbolic; actual flowcharts use standard geometric symbols.)

24.6 Steps to Create a Flowchart

1. **Define the Problem** – Understand what needs to be solved.
2. **Identify Inputs and Outputs** – What data enters and leaves the process.
3. **List Major Steps** – Break the process into logical operations.
4. **Arrange in Sequence** – Place steps in order of execution.
5. **Use Standard Symbols** – Represent each step with proper flowchart symbols.
6. **Connect with Arrows** – Show the flow of control or data.
7. **Validate** – Check logic, correctness, and completeness.

24.7 Example: Simple Addition Program Flowchart

Goal: Add two numbers and display the result.

Steps:

1. Start
2. Input A, B
3. Compute Sum = A + B
4. Display Sum
5. End

(Symbolically, this would use: *Start* → *Input* → *Process* → *Output* → *End*)

Unit. 5 MS-Office packages

25. Microsoft Office (MS Office)

Definition:
Microsoft Office (often called **MS Office**) is a **suite of productivity applications** developed by **Microsoft**. It includes programs for **word processing, data management, presentations, communication, and collaboration.**

First Release: 1988 (by Bill Gates at COMDEX)
Developer: Microsoft Corporation
Latest Version: Office 2024
Current Model: Subscription-based **Microsoft 365** (formerly Office 365)

25.1 Overview

- Originally included **Word, Excel, and PowerPoint** — still the **core applications** today.
- Has since expanded to include tools like **Outlook, Access, OneNote, and OneDrive.**
- Available on **Windows, macOS, mobile**, and **web platforms** (some components are Windows-only).
- **Microsoft 365** allows cloud-based access, collaboration, and regular feature updates.

25.2 Key Features of Microsoft Office

- **User Productivity:** Tools for writing, calculation, design, and data management.
- **Integration:** All applications are interoperable (e.g., linking Excel charts in Word).
- **Collaboration:** Real-time co-authoring via **Microsoft 365 and OneDrive**.
- **Extensibility:** Supports **VBA (Visual Basic for Applications)** for automation.
- **Cloud Services:** Files can be accessed anytime through **OneDrive integration**.

25.3 Core Applications

Application	Function / Purpose	Key Features
Microsoft Word	Word processor for creating and editing text documents.	Formatting tools, tables, mail merge, templates, PDF support.
Microsoft Excel	Spreadsheet software for data organization and calculations.	Formulas, charts, pivot tables, data analysis tools.
Microsoft PowerPoint	Presentation software for slideshows and visual communication.	Slide transitions, animations, templates, and media integration.
Microsoft OneNote	Digital notebook for note-taking and collaboration.	Organizes handwritten, typed, or multimedia notes; syncs via cloud.
Microsoft Outlook	Email and personal information manager.	Manages emails, calendars, contacts, and tasks.
Microsoft Access (Windows-only)	Database management system.	Combines GUI with relational database engine for data handling.

25.4 Additional and Specialized Applications

Application	Category	Use / Function
Publisher	Desktop Publishing	Used for brochures, cards, and newsletters (Windows-only).
Project	Project Management	Helps plan, track, and visualize project timelines.
Visio	Diagram & Flowcharting	Used for diagrams, organizational charts, and system designs.
Teams	Communication & Collaboration	Integrates chat, meetings, file sharing, and collaboration.
OneDrive	Cloud Storage	Syncs and stores files for remote access.
SharePoint	Server Collaboration Platform	Manages documents, intranets, and corporate portals.
Microsoft Forms	Online Survey Tool	Used for creating and analyzing forms and quizzes.
Microsoft Planner	Task Management	Used for organizing team tasks and projects visually.

25.5 Web and Cloud Services (Microsoft 365)

Service	Function
Microsoft Sway	Web-based presentation tool.
Delve	Organizes user data like emails, documents, and meetings.
Outlook.com / Outlook on the Web	Webmail platform similar to Outlook desktop.
Microsoft To Do	Task and reminder management app.
Microsoft Stream	Enterprise video-sharing service.
Microsoft Bookings	Appointment scheduling and booking app.

25.6 Mobile and Companion Apps

App	Function
Office Mobile	Unified mobile app (Word, Excel, PowerPoint) for Android/iOS.
Office Lens	Converts photos of documents or whiteboards into editable digital copies.
Office Remote	Turns a mobile phone into a remote control for presentations.

25.7 Server Applications

Server	Purpose
Microsoft Exchange Server	Mail and calendaring server.
Microsoft SharePoint Server	Document management and collaboration system.
Skype for Business Server	Real-time messaging and video conferencing.
Microsoft Project Server	Project and portfolio management.
Excel Services / InfoPath Forms Services	Web-based Excel and form management tools.

25.8 Evolution of Microsoft Office

Version	Key Highlight
Office 1.0 (1989)	Introduced Word, Excel, PowerPoint.
Office 97–2003	Added Outlook and Access.
Office 2007	Introduced the Ribbon interface.
Office 2013	Integrated with OneDrive and Cloud.
Office 2016 / 2019	Enhanced collaboration tools.
Office 365 (Now Microsoft 365)	Subscription model with cloud and AI integration.
Office 2024	Latest perpetual version.

List of MS Office Versions

Release date	Title	Details
1990-10-01	The Microsoft Office for Windows	Components: Word 1.1; Excel 2.0; PowerPoint 2.0.
1991-03-04	Office 1.5	Components: Word 1.1; Excel 3.0; PowerPoint 2.0.
1991-07-08	Office 1.6	Components: Word 1.1; Excel 3.0; PowerPoint 2.0; Mail 2.1. Note: Last version for Windows 3.0.
Early 1992	Office 2.0	Components: Word 2.0a; Excel 3.0; PowerPoint 2.0; Mail 2.1.
Mid 1992	Office 2.5	Components: Word 2.0a; Excel 4.0; PowerPoint 2.0e; Mail 2.1. Note: Mail 2.1 was licence only, no software.
1992-08-30	Office 3.0	Components: Word 2.0c; Excel 4.0; PowerPoint 3.0; Mail 3.0.
1994-01-17	Office 4.0	Components: Word 6.0; Excel 4.0; PowerPoint 3.0; Mail 3.1.
1994-06-02	Office 4.3	Components: Word 6.0; Excel 5.0; PowerPoint 4.0; Mail 3.2; Access 2.0. Notes: Final 16-bit version. Last for Windows 3.1x and Windows NT 3.1.
1994-07-03	Office for NT 4.2	Components: Word 6.0; Excel 5.0; PowerPoint 4.0; Office Manager. Notes: Office Manager included. Last version for Windows NT 3.5.
1995-08-24	Office 95 (7.0)	Components: Word, Excel, PowerPoint, Schedule+, Binder, Access, Bookshelf. Notes: First to use same version number (7.0) across major components; first fully 32-bit release.
1996-11-19	Office 97 (8.0)	Components: Word 97/98; Excel; PowerPoint; Access; Publisher 97/98; Outlook 97/98; BookShelf Basics; Small Business Financial Manager 97/98; plus add-ons. Notes: First version to receive extended support; shipped on CD and 45 floppy disks; Y2K-safe with SR2; last for Windows NT 3.51.
1997-06-20	Office 97 Powered by Word 98 (8.5)	Components: included Word 98 (JP/KR only). Notes: First to include Outlook 98 in all editions and Publisher 98 in Small Business Edition.
1999-06-07	Office 2000 (9.0)	Components: Word, Excel, Outlook, PowerPoint, Access, Publisher, Small Business Tools, FrontPage, PhotoDraw. Notes: First to receive 5 years extended support; final version without Product Activation and not covered by Office Genuine Advantage; last for Windows 95.

2001-05-31	Office XP (10.0)	Components: Word, Excel, Outlook, PowerPoint, Access, FrontPage, Publisher, Small Business Tools. Notes: Improved privilege-separation support; last for NT4.0, Win98, WinME; final 9x-based support; aka Office 2002; second product with 5 years extended support.
2003-08-19	Office 2003 (11.0)	Components: Word, Excel, Outlook, PowerPoint, Access, Publisher, OneNote, InfoPath. Notes: First to support NT-only; introduced OneNote; final legacy interface; last for Windows 2000.
2007-01-30	Office 2007 (12.0)	Components: Word, Excel, PowerPoint, Outlook, Publisher, Access, Groove, OneNote, InfoPath, Communicator, Visio Viewer, OCT. Notes: First to use the Ribbon UI; Calibri becomes default font; continued extended support lifecycle.
2010-06-15	Office 2010 (14.0)	Components: Word, Excel, PowerPoint, OneNote, Outlook, Publisher, Access, InfoPath, SharePoint Workspace, Visio Viewer, OCT, Lync. Notes: First 32-bit & 64-bit shipping; last for Windows XP/Server 2003/Vista/Server 2008. (Version 13 skipped.)
2013-01-29	Office 2013 (15.0)	Components: Word, Excel, PowerPoint, OneNote, Outlook, Publisher, Access, Lync (later Skype for Business), Visio Viewer. Notes: Lync replaced by Skype for Business after update.
2015-09-22	Office 2016 (16.0)	Components: Word, Excel, PowerPoint, OneNote, Outlook, Publisher, Access, Skype for Business, Visio Viewer. Notes: Final Office to receive 5-year mainstream support; last for many legacy Windows versions (Win7/Server2008R2/8/8.1/Server2012/2016).
2018-09-24	Office 2019 (16.0)	Components: Word, Excel, PowerPoint, OneNote, Outlook, Publisher, Access, Skype for Business, Visio Viewer. Notes: Final version to receive extended support; second perpetual Office 16 release adding some Office 365 features; OneNote removed (UWP bundled with Windows), x86 OneNote returned Mar 2020.
2021-10-05	Office 2021 (16.0)	Components: Word, Excel, PowerPoint, OneNote, Outlook, Publisher, Access, Skype for Business, Visio Viewer. Notes: First to receive 3 years extended support; final version with Calibri as default font.
2024-10-01	Office 2024 (16.0)	Components: Word, Excel, PowerPoint, OneNote, Outlook, Access, Skype for Business, Visio Viewer. Notes: First to set Aptos as the default font across all apps; second perpetual release to receive 3 years extended support; Current release.

Unit : 6 DBMS Package

27. Database Management System (DBMS)

Definition:
A **Database Management System (DBMS)** is **software** that helps **define, manipulate, retrieve, and manage data** in databases.
It acts as a bridge between **databases, applications, and users**, ensuring data consistency, security, and accessibility.

27.1 Functions of a DBMS

A DBMS performs several essential tasks to maintain and manage data effectively:

1. **Defining Rules** – Establishes constraints and relationships to validate and organize data.
2. **Data Manipulation** – Allows inserting, updating, deleting, and retrieving data.
3. **Interaction Management** – Connects users and applications with the database.
4. **Data Analysis** – Supports queries and reports for decision-making.
5. **Storage & Retrieval** – Efficiently saves and fetches information from data files.

27.2 Key Roles of a DBMS

- Maintains **data integrity** (accuracy and consistency).
- Controls **access and security** of information.
- Enables **multi-user access** and concurrent transactions.
- Provides **backup and recovery** mechanisms.
- Simplifies **application development** through APIs and SQL support.

27.3 Popular Database Management Systems

DBMS	Type	Developer / Origin	Latest Version	License	Highlights / Usage
MySQL	RDBMS	Oracle Corporation (originally MySQL AB)	8.0.32	GNU GPL v2 / Proprietary	Open-source, widely used in websites like WordPress, Twitter, Facebook.
MariaDB	RDBMS	MariaDB Foundation / MariaDB Corporation Ab	11.1.0	GPL v2	Fork of MySQL, free and community-driven; used by Wikipedia, Google.
Microsoft SQL Server	RDBMS	Microsoft	SQL Server 2022	Proprietary	Commercial DBMS for enterprises; supports Windows and Azure.
Oracle Database (Oracle DBMS)	Multi-model DBMS	Oracle Corporation	23c (beta), 19c (LTS)	Proprietary	Used for OLTP and Data Warehousing; enterprise-grade system.

PostgreSQL	RDBMS	PostgreSQL Global Development Group	15.2	PostgreSQL License	Open-source, ACID-compliant, supports advanced features (JSON, GIS).
MongoDB	NoSQL (Document-Oriented)	MongoDB Inc.	6.0.4	SSPL	Stores data as JSON-like documents; used for big data and cloud apps.
Redis	NoSQL (Key-Value Store)	Redis (by Salvatore Sanfilippo)	7.0	BSD 3-Clause	High-speed, in-memory data store used for caching and session data.
IBM DB2	RDBMS	IBM	11.5.8	Proprietary	Supports large-scale enterprise databases; used in banking and analytics.
Elasticsearch	NoSQL (Search & Analytics Engine)	Elastic NV (by Shay Banon)	8.7	Elastic License / SSPL	Distributed search engine; used for logs, analytics, and full-text search.
SQLite	Embedded RDBMS	Dwayne Richard Hipp	3.41.2	Public Domain	Lightweight and self-contained; used in mobile apps and browsers.

27.4 Classification of DBMS by Data Model

Type	Description	Examples
Relational DBMS (RDBMS)	Stores data in tables (rows & columns).	MySQL, Oracle, PostgreSQL
Object-Oriented DBMS (OODBMS)	Stores data as objects.	ObjectDB, db4o
Object-Relational DBMS (ORDBMS)	Combines relational and object features.	PostgreSQL
NoSQL DBMS	Stores non-relational, unstructured data.	MongoDB, Redis, Cassandra
Hierarchical DBMS	Organizes data in tree-like structures.	IBM IMS
Network DBMS	Uses complex relationships among records.	IDMS, Raima Database

27.5 Key Components of a DBMS

- **Database Engine:** Core part managing storage, retrieval, and transactions.
- **Data Dictionary:** Stores metadata (schema details, field types, etc.).
- **Query Processor:** Interprets and executes database queries.
- **Transaction Manager:** Ensures consistency during concurrent operations.

- **Authorization Manager:** Handles user permissions and security.
- **Utilities:** Tools for import/export, backup, and tuning.

27.6 Common Database Languages

Language	Purpose	Example Command
DDL (Data Definition Language)	Defines structure and schema.	CREATE TABLE Students (...);
DML (Data Manipulation Language)	Manipulates data.	INSERT INTO Students VALUES (...);
DQL (Data Query Language)	Retrieves data using queries.	SELECT * FROM Students;
DCL (Data Control Language)	Manages access rights.	GRANT SELECT ON Students TO User1;

27.7 DBMS Interaction and APIs

- **Database Applications:** Web tools, dashboards, or custom apps that access data.
- **APIs:**
  - **ODBC** – Open Database Connectivity (universal standard).
  - **JDBC** – Java Database Connectivity.
  - **ADO.NET** – Microsoft's data access technology.

28. FoxPro (Database Management System)

Definition:
FoxPro is a **text-based, procedurally oriented programming language and database management system (DBMS)** that later evolved to support **object-oriented programming**.
It was developed by **Fox Software**, later acquired and maintained by **Microsoft**.

28.1 Overview

- **Type:** Procedural and Object-Oriented DBMS
- **Initial Developer:** Fox Software (Perrysburg, Ohio)
- **Later Developer:** Microsoft (after 1992 acquisition)
- **Supported Platforms:** MS-DOS, Windows, Macintosh, UNIX
- **Latest Version:** FoxPro 2.6
- **Successor: Visual FoxPro** (discontinued in 2007)

28.2 Origin and Evolution

Software	Derived From	Remarks
dBase II	—	Original database system by Ashton-Tate.
dBase III	dBase II	Improved relational data handling.
FoxBase	dBase III	Created by Fox Software; faster and more user-friendly.
FoxPro	FoxBase	Added enhanced indexing, query optimization, and relational features.

28.3 Nature of FoxPro

- A combination of **DBMS** and **RDBMS** (supports relations between multiple .dbf files).
- **Text-based** and **menu-driven** interface.
- **Procedural language** with optional **object-oriented features**.
- Does **not support transactional processing** (a limitation).
- Used for **rapid application development** and **data handling automation**.

28.4 Technical Features

Feature	Description
Rushmore Technology	Query optimization engine that speeds up data retrieval using indexed fields.
Indexing	Supports multiple indexes for faster search and sorting operations.
Compatibility	Operates on multiple OS platforms like MS-DOS, Windows, Macintosh, UNIX, and Linux (via ibcs2).
Development Environment	Built using Watcom C/C++ with DOS/4GW memory extender.
Memory Management	Could access all available RAM even without HIMEM.SYS loaded.

28.5 Operating System Compatibility

Version	MS-DOS	Windows (3.1–7)	Macintosh	UNIX / Linux / FreeBSD	Windows 2000–10 (32-bit)
FoxPro 2.0	Yes	Yes	Yes	Yes	No
FoxPro 2.5	Yes	Yes	Yes	Yes	No
FoxPro 2.6	Yes	Yes	Yes	Yes	Yes

28.6 Reception and Popularity

- **Praised** for its **speed**, **ease of use**, and **rapid application development** capability.
- Users could create **database browsers with interfaces in under 100 lines of code**.
- **Tom Campbell (Compute!, 1993):** Highlighted its value and efficiency for professional developers.
- **Richard O'Reilly (Los Angeles Times):** Considered FoxPro 2.5 faster and easier to use than Microsoft Access and Borland Paradox.
- **Wayne Ratliff (Creator of dBase):** Admired FoxPro's structured and "friendly" design.

28.7 Strengths of FoxPro

- Fast query execution using **Rushmore optimization**.
- Excellent **relational data handling** for its time.
- Cross-platform support (DOS, Windows, Mac, UNIX).
- Provided **powerful indexing and data manipulation** tools.
- Easy to develop **menu-driven and form-based applications**.

28.8 Limitations

- **No transactional processing** (can't guarantee data recovery after system failure).
- **Limited object-orientation** (before Visual FoxPro).
- **Outdated GUI** and text-based interface in early versions.
- **Discontinued support** after 2007 by Microsoft.

28.9 Visual FoxPro

- Successor to FoxPro, introduced full **object-oriented programming** and GUI development.
- Integrated better with **Windows APIs**.
- Officially discontinued in **2007**, marking the end of FoxPro's evolution.

29. CDS/ISIS (Computerised Documentation Service / Integrated Set of Information Systems)

Definition:
CDS/ISIS is a **software package for generalized Information Storage and Retrieval (ISR) systems**, developed, maintained, and distributed by **UNESCO**.
It is especially designed for **bibliographic and library applications**, widely used by **small and medium-sized libraries** for cataloguing and database management.

29.1 Developer and Origin

Aspect	Details
Developed by	UNESCO (United Nations Educational, Scientific and Cultural Organization)
Initial Design	Mid-1970s by *Mr. Giampaolo Del Bigio* for UNESCO's Computerized Documentation System (CDS)
First Release	1985
Platform	Initially on IBM Mainframe, later on Mini and Microcomputers
Programming Language	Originally in Pascal
License	Free for non-commercial use (UNESCO license)
Languages Available	Arabic, Chinese, English, French, German, Portuguese, Russian, Spanish, etc.

29.2 Meaning of CDS/ISIS

- **CDS:** Computerised Documentation Service
- **ISIS:** Integrated Set of Information Systems

Together, they represent a **complete system for managing and retrieving structured information** in libraries and documentation centers.

29.3 Features

- Designed for **information storage, retrieval, and dissemination**.
- Suitable for **bibliographic databases** and **library cataloguing**.
- Handles **structured and textual data** efficiently.
- Supports **multi-lingual interfaces and Unicode (in newer versions)**.

- Offers **customizable data entry, search, and display formats**.
- **Free for educational and non-commercial library applications**.

29.4 Versions and Developments

Version / Tool	Year / Platform	Description / Features
Mainframe CDS/ISIS	1970s	Original version for IBM mainframe systems.
Micro CDS/ISIS	1985	Pascal-based version for IBM PCs under MS-DOS.
WinISIS	1995	Windows version; supports LAN installations and GUI interface.
JavaISIS	2000	Client/server component enabling remote access via Internet for Windows, Linux, and macOS.
GenISIS	Early 2000s	Tool for generating HTML web forms for online database searching.
ISIS_DLL	2000s	API allowing programmers to develop CDS/ISIS-based applications.
OpenIsis	2002–2004	Independently developed open-source API compatible with CDS/ISIS databases.
J-Isis	2005 onwards	Modern Unicode-based FOSS version maintained by UNESCO (developer: Jean Claude Dauphin).

29.5 Key Components and Tools

- **Database Definition (FDT):** Defines field structure and data types.
- **Data Entry Worksheets (FMT):** Templates for user data entry.
- **Display Formats (PFT):** Control how information is displayed or printed.
- **Inverted File (IFP):** Index file enabling fast retrieval.
- **Search Engine:** Allows boolean, keyword, and field-based searching.

29.6 Advantages of CDS/ISIS

- **Free and globally distributed** software (ideal for developing countries).
- Highly **customizable** to suit library or documentation needs.
- Supports **multi-user** and **multi-database** environments.
- Available in **multiple languages** and adaptable to local needs.
- **Efficient for bibliographic data** and cataloguing systems.
- **Lightweight** and can run on low-resource systems.

29.7 Limitations

- **Complex configuration** for beginners.
- **Outdated interface** in older versions (WinISIS).
- **Limited multimedia handling** compared to modern systems.
- Requires **technical knowledge** for advanced customization.

29.8 Applications in Libraries

- **Library catalogues and OPACs (Online Public Access Catalogues)**.
- **Indexing and abstracting databases.**
- **Bibliographic information systems.**
- **Archival and documentation management.**
- **Research and institutional repositories.**

29.9 Current Development: J-Isis

- **Platform:** Cross-platform (Windows, Linux, macOS).
- **Language:** Java-based (Unicode-compliant).
- **Nature:** Fully Free and Open Source Software (FOSS).
- **Developer:** UNESCO, maintained by *Jean Claude Dauphin*.
- **Goal:** Modernize CDS/ISIS into a web-compatible, open, and extensible tool.

29.10 Significance

- CDS/ISIS is regarded as the **standard library software in developing countries**.
- It laid the **foundation for later integrated library systems (ILS)**.
- Promoted **information access, cataloguing, and knowledge sharing** worldwide under UNESCO's vision.

30. SOUL (Software for University Libraries)

Definition:
SOUL is a **state-of-the-art Integrated Library Management Software (ILMS)** developed by the **INFLIBNET Centre** to automate all major housekeeping operations in **college and university libraries**. It is **user-friendly**, **network-compatible**, and designed to function in a **client–server environment**.

30.1 Developer and Background

Aspect	Details
Full Form	Software for University Libraries
Developed by	INFLIBNET Centre (Information and Library Network Centre), Gandhinagar, India
Initial Release	SOUL 1.0 – during CALIBER 2000
Later Versions	SOUL 2.0 – January 2009; SOUL 3.0 – February 2021
Type	Integrated Library Management System (ILMS)
Target Users	University, college, and school libraries
Environment	Client–Server
Database Support	MS-SQL, MySQL, or other RDBMS platforms

30.2 Objectives of SOUL

- To **automate library housekeeping operations**.
- To **standardize bibliographic records** and data exchange.

- To **support multilingual data entry** using Unicode.
- To provide a **network-enabled library system** suitable for all sizes of libraries.
- To promote **resource sharing and networking** among academic institutions.

30.3 Key Features

Feature	Description
User-Friendly Interface	Simple, menu-driven system with easy navigation.
Client–Server Model	Ensures faster access and multi-user capability.
Standards Compliance	Supports **MARC 21**, **Unicode (UCS)**, **NCIP 2.0**, and **SIP 2** protocols.
Multilingual Support	Handles Indian and international languages efficiently.
Customizable Modules	Libraries can adapt features as per their workflow.
Web Integration	Compatible with online catalogs and digital library links.
Security and Backup	Includes user-level access control and data backup tools.

30.4 Versions of SOUL

Version	Year Released	Major Enhancements
SOUL 1.0	2000	Initial version with basic automation features for university libraries.
SOUL 2.0	2009	Added Unicode support, MARC 21 compliance, and international networking protocols.
SOUL 3.0	2021	Improved GUI, integration with modern RDBMS, enhanced web compatibility, and cloud readiness.

30.5 Major Modules in SOUL

1. **Acquisition Module** – Handles book and journal orders, invoices, and budget control.
2. **Cataloguing Module** – Provides MARC 21–based bibliographic data entry and management.
3. **Circulation Module** – Manages issue, return, renewal, fine collection, and membership.
4. **Serial Control Module** – Maintains records of journals and periodicals.
5. **OPAC (Online Public Access Catalogue)** – Enables users to search resources through web-based interfaces.
6. **Administration Module** – User management, database backup, and system configuration.

30.6 Technical Specifications

- **Platform:** Windows/Linux (Server and Client)
- **Database:** MS-SQL, MySQL, or compatible RDBMS
- **Networking:** Supports LAN and internet-based access
- **Protocols Supported:**
    - **MARC 21** – for bibliographic records
    - **Unicode (UCS)** – for multilingual support

- **NCIP 2.0 / SIP 2** – for circulation and access control

- **Security:** User authentication and role-based permissions

31. MS Access – A GUI Database Management System

Definition:
Microsoft Access (MS Access) is a Graphical User Interface (GUI)–based Relational Database Management System (RDBMS) developed by Microsoft in 1992.
It is a part of the Microsoft Office Suite and combines a database engine, graphical tools, and software development features, allowing users to create and manage databases easily without extensive programming knowledge.

31.1 Basic Information

Aspect	Details
Developer	Microsoft Corporation
First Released	1992
Latest Version	MS Access 2021
Included In	Microsoft Office Suite
Database Engine	ACE (Access Database Engine), formerly Jet Engine
Type	GUI-based Relational Database Management System
Interface	Graphical (Forms, Tables, Reports, Macros)
Platform	Windows OS

31.2 Characteristics

- **User-Friendly Interface:** No coding expertise needed to manage data.
- **GUI-Based:** Entirely visual — tables, forms, queries, and reports.
- **Relational Structure:** Stores data in **tables (rows and columns)** with relationships between them.
- **Part of MS Office:** Works seamlessly with Excel, Word, and Outlook.
- **Customizable Applications:** Allows front-end creation using **Forms, Macros, and VBA**.
- **Supports SQL:** Queries can be created visually or through SQL commands.

31.3 Features of MS Access

Feature	Description
Primary Key Definition	Users can define a primary key to uniquely identify records.
Table, Query, Form & Report Creation	Enables creation of relational tables, queries for retrieval, forms for data entry, and reports for presentation.
Import/Export Options	Supports data transfer from Excel, SQL Server, Oracle, FoxPro, HTML, XML, etc.
Templates	Predefined templates simplify database creation.
Multi-Valued Fields	Allows storing more than one value in a single field.
Reporting Tools	Generates custom reports with charts and summaries.
Macros & VBA	Automates repetitive tasks and adds logic-based functionalities.
"Tell Me What to Do" Feature	Provides quick help for creating database components.

Relational Linking	Establishes and maintains relationships between multiple tables.

31.4 Components of MS Access

MS Access consists of **seven key components**, each serving a distinct purpose in database design and management:

1. **Tables:**
    - Store data in rows and columns.
    - Fundamental structure of the database.
    - Each table has defined data types and a primary key.
2. **Forms:**
    - User interfaces for entering or viewing data.
    - Can be **bound** (connected to tables) or **unbound** (standalone).
3. **Reports:**
    - Used for **data presentation and analysis**.
    - Summarizes and formats data for printing or sharing.
4. **Relationships:**
    - Define connections between tables for data consistency.
    - **Types:** One-to-One, One-to-Many, Many-to-One, Many-to-Many.
5. **Queries:**
    - Commands to **retrieve, modify, or insert** data.
    - Can be created graphically or through **SQL statements**.
6. **Macros:**
    - Predefined instructions for **automating repetitive tasks**.
    - Example: Auto-generating reports or opening forms.
7. **Modules:**
    - Blocks of **VBA (Visual Basic for Applications)** code.
    - Used to perform **custom operations** and enhance functionality.

31.5 Uses of MS Access

- **Data Management:** Store, retrieve, and organize large volumes of data.
- **Application Development:** Used for creating **desktop and web-based applications**.
- **Integration:** Imports/exports data to/from **Excel, SQL Server, Oracle, and Outlook**.
- **Education and Research:** Used in schools and universities for **record management and scheduling**.
- **Front-End Design:** Acts as a **front-end interface** for larger databases (e.g., SQL Server).

- **Automation:** Automates repetitive operations using **Macros and VBA modules**.

Unit : 7 Computer application to library & Information work

32. History of ICT / Automation Application in Library in India

Definition:
The **application of Information and Communication Technology (ICT)** in Indian libraries refers to the **use of computers, networks, and digital systems** to automate library operations such as cataloguing, circulation, and information retrieval.
Library automation in India evolved gradually from the **1950s**, beginning with basic mechanization to the modern era of **digital and networked libraries**.

32.1 Early Beginnings (1950s–1960s)

Year/Period	Developments
1955	The **Indian Statistical Institute (ISI), Calcutta**, became the first institution in India to install a **computer system**.
1960s	Early attempts were made by libraries and information centers to automate processes using **punch cards** and **second-generation computers** at IIT Kanpur and IIT Bombay.
1964	ISI developed an **indigenous computer**, marking a significant milestone for computing in India.
1964–1965	**INSDOC (Indian National Scientific Documentation Centre)** began experiments with computers for **documentation and information work** using the **IBM 1620 Model I** available at IIT Kanpur.
1965	INSDOC computerized **author and subject indexes** of *Indian Science Abstracts* — the **first library automation effort in India** (Raizada, 1965).

32.2 Expansion of Experiments (Late 1960s–1970s)

Year	Event / Contribution
1967	INSDOC produced the *Roster of Indian Scientific and Technical Translators* using computers — a pioneering bibliographic automation effort.
1970s	More libraries began preparing **computerized databases** for bibliographic and research purposes.
1978	INSDOC introduced **Selective Dissemination of Information (SDI)** services as part of a **NISSAT** project, using **CAN/SDI software** from IIT Madras and databases like **Chemical Abstracts** and **INSPEC**.

32.3 Growth through Networking (1980s–1990s)

Decade	Key Developments
1980s	Introduction of **microcomputers, optical disks, and CD-ROMs** in libraries. Increased use of **mini and microcomputers** for automation.
Late 1980s–1990s	Major government-backed programs began promoting **library networks** through **NISSAT** and **INSDOC**.
Key Library Networks Established:	

- **CALIBNET (Calcutta Library Network)**
- **DELNET (Delhi Libraries Network)**

- **INFLIBNET (Information and Library Network)**
- **PUNENET (Pune Library Network)**
- **NICNET, INDONET, SIRNET** — other government and research-based networks. |

32.4 Institutional Role in Library Automation

Institution	Contribution
INSDOC (Now NISCAIR/CSIR-NIScPR)	Early automation in documentation and abstracting services.
NISSAT (National Information System for Science and Technology)	Financial and technical support for library networks.
DRTC (Documentation Research and Training Centre)	Training and research in library computer applications.
DESIDOC (Defence Scientific Information and Documentation Centre)	Developed software and systems for scientific libraries.
INFLIBNET (Information and Library Network Centre)	Provided nationwide library automation and networking support to universities and research institutions.

32.5 Establishment of INFLIBNET – A Turning Point

- Founded in **1991** under **UGC (University Grants Commission)**.
- Served as a **catalyst for library automation in academic institutions**.
- Provided **funding, standards, and training** for automation projects.
- Encouraged **inter-library networking and resource sharing** through digital catalogues and online databases.
- Before INFLIBNET, only select institutions like **IITs and IIMs** were experimenting with automation.

(Singh, 2003: "INFLIBNET proved a real catalyst by providing finance and laying standards.")

32.6 Growth and Development Timeline (Kaul, 1999)

Period	Developments in Library Automation
1940–1949	Semi-mechanical systems like **edge-notched cards**, **optical coincidence**, and **peek-a-boo cards**.
1950–1959	Introduction of **punched cards**, **data processing machines**, and **micro-image searching systems**.
1960–1969	Use of **digital computers**, **online systems (feasibility studies)**, and **experiments in library networking**.
1970–1979	Design of **online systems**, **conversion of batch to online**, and **growth of library networks**.
1980–1989	**Intensive automation**, **use of mini/microcomputers**, **optical disks**, **CD-ROMs**, and **FAX**.

33. Areas and Functions for ICT / Automation in a Library

Definition:
The use of Information and Communication Technology (ICT) in libraries enhances efficiency, accuracy, and accessibility by automating traditional library operations.
Automation reduces manual effort and helps manage library resources digitally — from book ordering to information dissemination.

33.1 Major Areas of ICT Application in Libraries

ICT is applied across all **housekeeping and service functions** of a library, including **acquisition, cataloguing, circulation, serial control, management, reference services**, and **user access**.

1. Ordering and Acquisitions

Purpose:
To automate the **selection, ordering, and accession** of new library materials.

Functions Automated:

- Ordering and receipting of materials
- Preparation of **acquisition and accession lists**
- Maintenance of **order files and reports**
- **Claiming** of delayed or missing items
- **Fund accounting** and budget control
- Enquiries about **order status**
- Statistical and financial **reports generation**

Example:
Automated acquisition in **SOUL** or **KOHA** enables librarians to generate purchase orders and update book status instantly.

2. Serials Management

Purpose:
To control and track **periodicals, journals, and magazines** using ICT tools.

Functions Automated:

- Subscription ordering and renewals
- Receipting of individual issues
- Claiming of missing issues
- **Serials check-in/check-out** management
- Creation of **union catalogues** and **holding lists**
- **Binding control** for completed volumes
- **Fund accounting** for serial expenditures
- **Cataloguing** of serial titles

- **Circulation control** of serials
- Generating **inquiries, reports, and statistics**

Example:
DELNET and INFLIBNET use automated serials modules to maintain journal holdings across libraries.

3. Cataloguing and Classification

Purpose:
To create, organize, and maintain **bibliographic and authority records** efficiently.

Functions Automated:

- **Data entry** of bibliographic details
- **Authority control** for consistent headings
- **Downloading of MARC records** from external databases (like OCLC, OPACs)
- Creation of **Online Public Access Catalogues (OPACs)**
- Access for **remote users** via the Internet
- **Catalogue card and spine label production**
- **Retrospective conversion** of old manual records

Example:
Using **MARC 21** format in **SOUL** or **LibSys** for cataloguing and classification.

4. Circulation Control

Purpose:
To manage the **issue, return, and renewal** of documents to users.

Functions Automated:

- Setting **loan policies** (loan period, fines, renewal limits)
- **Issue/Return** of books and materials
- **Renewals and reservations** of items
- **Fine calculation** and payment tracking
- **Borrower database maintenance**
- **Barcode/RFID integration** for transaction speed
- Enquiries on borrower history and book status
- Generation of **notices**, **reminders**, and **statistical reports**

Example:
RFID-enabled circulation in university libraries automates issue-return through self-check systems.

5. Library Management

Purpose:
To aid in the **administrative and financial** functions of a library.

Functions Automated:

- **Budgeting and accounting** operations
- **Word processing** and official correspondence
- **Scheduling, planning**, and resource allocation
- **Statistical analysis and report generation**
- Tools for **decision-making and performance evaluation**

Example:
Libraries use MS Office, SOUL, and ERP-based modules for financial and administrative management.

6. Inter-Library Loan (ILL) Services

Purpose:
To enable **document exchange and resource sharing** among libraries.

Functions Automated:

- Entry of ILL requests
- Issue and return tracking for borrowed/lent items
- Fine and overdue management
- Borrower file maintenance
- Enquiries and status updates
- Reports and statistics on interlibrary transactions

Example:
DELNET and INFLIBNET automate ILL requests through online forms and catalog sharing.

7. Audio-Visual Management

Purpose:
To manage **non-print materials** (CDs, DVDs, tapes, digital videos, etc.).

Functions Automated:

- Acquisition and cataloguing of audio-visual items
- Storage, retrieval, and metadata tagging

Example:
Multimedia cataloguing in **SOUL** or **KOHA** with linked file attachments.

8. Information Storage and Retrieval

Purpose:
To manage digital databases and enable efficient **searching and access** to information.

Functions Automated:

- **Database creation and management**
- **Online database searching** using Boolean logic

- **Downloading and uploading** of information
- **Indexing and abstracting** of documents

Example:
Online bibliographic databases (like Scopus, JSTOR) use advanced search interfaces and metadata indexing.

9. Reference and Information Services

Purpose:
To assist users in **information discovery and literacy**.

Functions Automated:

- Creation of **bibliographic listings and guides**
- **Online reference assistance** via email or chat systems
- **User education and library instruction** modules
- **Public access to databases and OPACs**

Example:
Digital reference desks in academic libraries using **LibAnswers** or **Ask-a-Librarian** tools.

10. Community Information Services

Purpose:
To provide access to **local or community-based data and literacy programs**.

Functions Automated:

- **Data entry** of community records
- **Online public access** to community databases
- **Public literacy programs** using ICT
- **Public access interfaces** for community learning

Example:
Public libraries providing **digital access points** and **literacy initiatives** under the **National Digital Library Mission (NDLI)**.

33.2 Summary Table

Area of Application	Key ICT/Automation Functions
Acquisition	Ordering, fund accounting, claiming, accession listing
Serials Control	Subscription, check-in/out, binding, reporting
Cataloguing	MARC records, OPAC, authority control
Circulation	Issue-return, renewals, fines, borrower data
Management	Accounting, budgeting, statistics
ILL Services	Document exchange, status reports
Audio-Visuals	Cataloguing and retrieval of non-print media
Information Retrieval	Database creation, online search, indexing
Reference Services	Bibliographic listings, user assistance
Community Information	Public data access, literacy initiatives

Unit : 8 Communication Technology

34. Functions and Means of Communication

34.1 Functions of Communication

Communication plays a vital role in individual and social life.
According to **MacBride (1980)**, communication performs several **socio-psychological functions** within a social system.

Function	Explanation
1. Information	Communication enables **collection, storage, and dissemination** of information for **public understanding and decision-making**. It acts as a **resource for social, cultural, and economic development**.
2. Socialisation	Helps individuals **learn social norms** and become active members of society. Through communication, people **understand and respect others' ideas, emotions, and values**.
3. Motivation	Encourages individuals and communities to **work toward shared goals**. Well-informed citizens are more motivated to **participate in social welfare and development activities**.
4. Education	Communication spreads knowledge and **enhances intellectual and skill development**. The quality of education depends on **effective communication between educators and learners**.
5. Entertainment	Provides **relaxation and enjoyment** through dance, drama, music, sports, and other cultural forms. Entertainment communication supports **emotional well-being and social unity**.

Summary:
Communication informs, connects, motivates, educates, and entertains — forming the backbone of **social interaction and human progress**.

34.2 Means of Communication

Communication operates through a variety of **tools, media, and technologies** that help collect, produce, store, and share information.

Means	Description / Role
1. Signs and Sounds	The **earliest form** of communication — includes body language, gestures, facial expressions, drawings, paintings, songs, and music. Still used in many **tribal and folk traditions**.
2. Language	The **most powerful human tool** for sharing ideas and emotions. India alone has **22 scheduled languages** and hundreds of dialects. Language enables **interpersonal, national, and international communication**, though linguistic diversity can also create barriers.
3. Postal System	The **oldest organized network** for long-distance message delivery. Remains essential for **official, educational, and institutional communication**, including correspondence in **open and distance education** (e.g., IGNOU).
4. Telephone	Enables **real-time interpersonal communication**. Though underused in education earlier, **mobile phones and wireless networks** have made telecommunication widespread in India, transforming **interactive and distance learning**.

5. Mass Media	Includes **printing press, radio, television, newspapers, and books**. These media reach large audiences, aiding **mass education, public awareness, and social change**. Modern broadcasting transcends geographic boundaries.
6. Satellites	Revolutionized global communication by enabling **instant data transmission** across vast distances. India's **INSAT system** supports **television, radio, weather forecasting, remote sensing, and educational telecasts**.
7. Computers	Represent the **most advanced communication technology**, enabling **interactive, digital, and personalized communication**. Tools like **email, the Internet, and online platforms** have transformed how people **learn, share, and collaborate**.

35. Process of Communication

35.1 Introduction

The process of communication refers to the sequence of steps through which a message travels from the sender (source) to the receiver (destination).
The most accepted model is that of Shannon and Weaver (1949), which identifies key elements — source, message, channel, receiver, feedback, and noise.

Communication is not a one-way act; it's a continuous and cyclical process involving sending, receiving, and responding.

35.2 Major Elements in the Communication Process

Element	Description / Role
1. Sender (Source)	The **originator of the message** — an individual, organization, document, or medium that begins communication. The **credibility of the sender** affects how the receiver interprets the message.
2. Idea / Thought	The **concept, fact, or emotion** the sender wishes to communicate. Communication begins only when there is something meaningful to express.
3. Encoding	The process of **converting ideas into symbols** (words, pictures, gestures, or sounds). The sender chooses **language and structure** suitable for the medium and receiver. Example: The style of a telegram differs from a face-to-face conversation.
4. Conceptual Filters	The sender's **knowledge, attitude, experience, values, and psychological traits** influence how a message is encoded. These filters can shape or distort meaning. Example: A biased opinion can affect how information is presented.
5. Transmission	The act of **sending the encoded message** through a selected **channel or medium** (verbal, written, electronic, or non-verbal).
6. Message	The **content or information** that the sender wants to share. It can be in **verbal, written, visual, or symbolic** form.
7. Channel	The **path or route** through which the message travels — e.g., air (for speech), paper (for letters), cables, or digital networks.
8. Medium	The **carrier or tool** used to send the message — such as **telephone, email, newspaper, radio, or television**. The medium can be **verbal (oral/written)** or **non-verbal** (gestures, symbols, body language).

9. Receiver	The **intended audience or target** who interprets the message. The receiver decodes the signal using their own **knowledge, perception, and experience**. Effective communication depends on how accurately the receiver understands the sender's meaning.
10. Decoding	The process by which the receiver **interprets or translates** the encoded symbols into meaningful understanding.
11. Feedback	The **response** or **reaction** from the receiver to the sender. Feedback closes the communication loop, ensuring that the message has been received and understood correctly. Example: A nod, reply, or question from the listener.
12. Noise	Any **disturbance or interference** that distorts or interrupts message transmission. Noise can be **physical (sound, poor signal)**, **psychological (bias, inattention)**, or **semantic (language confusion)**.

35.3 Schematic Representation (Conceptual Summary)

Sender → Encoding → Message → Channel/Medium → Receiver → Decoding → Feedback (with possible Noise at any stage)

(Adapted from Shannon and Weaver, 1949; Hoy & Miskel, 1991)

35.4 Explanation of Key Stages

1. **Idea Formation** – The sender decides what to communicate.
2. **Encoding** – The idea is expressed in symbols, words, or visuals.
3. **Transmission** – The message is sent through a medium.
4. **Reception and Decoding** – The receiver interprets it.
5. **Feedback** – The receiver's response reaches the sender.
6. **Noise** – Any interference that alters or hinders understanding.

35.5 Example

Situation: A librarian sends an email to students about new arrivals.

- **Sender:** Librarian
- **Message:** Notice of new books
- **Encoding:** Typed email with proper wording
- **Channel/Medium:** Internet (email system)
- **Receiver:** Students
- **Decoding:** Reading and understanding the message
- **Feedback:** Students respond or visit the library
- **Noise:** Email delivery delay, spam filter, or misinterpretation

35.6 Importance of Feedback in Communication

- Confirms **message understanding**
- Enables **correction of errors or confusion**
- Makes communication **interactive and dynamic**
- Builds **trust and engagement** between participants

35.7 Role of Noise

Noise can occur **at any stage** and reduce clarity or accuracy.
Types of Noise:

- **Physical:** External disturbances like loud sounds or poor signal.
- **Psychological:** Emotions, stress, or bias.
- **Semantic:** Misunderstanding due to language or terminology.
- **Technical:** Faults in equipment or media failure.

36. Types of Communication

36.1 Introduction

Communication operates at multiple levels depending on the **number of participants**, **context**, and **purpose**.
Broadly, it can be classified into **four main types**:

1. **Interpersonal Communication**
2. **Group Communication**
3. **Organisational Communication**
4. **Mass Communication**

36.2 Types of Communication

Type	Description / Features	Examples
1. Interpersonal Communication	- The **most basic and direct form** of communication between **two individuals**. - Involves **exchange of ideas, emotions, and thoughts**. - Can occur **face-to-face or through a medium** like a phone or video call. - Builds **understanding and personal connection** between sender and receiver.	- Conversation between friends - Teacher and student interaction - Mother and child discussion - Telephone conversation
2. Group Communication	- Takes place within a **group of people** who share a **common interest or goal**. - Involves **discussion, debate, and decision-making** among members. - Promotes **cooperation, problem-solving, and social bonding**. - Usually **interactive** and participatory.	- Family discussions - Seminar or workshop sessions - Office team meetings - Classroom group activities
3. Organisational Communication	- Refers to communication that takes place **within an organisation**. - Includes both **formal** (official notices, memos, reports) and **informal** (casual talks, feedback) communication. - Ensures **coordination and information flow** among various departments. - May move in **upward, downward, or horizontal** directions within the system. - Involves multiple group communications within a structured setup.	- Office circulars and notices - Emails within departments - Staff meetings in a company - Internal communication in universities or libraries
4. Mass Communication	- Communication that targets a **large, diverse, and widespread audience**. - Conducted through **mass media** such as newspapers, radio, television, or the Internet. - Purpose: to **inform, educate, or entertain** the public. - Involves **institutional sources** and **technological means** to reproduce and transmit information. - The audience is **large, anonymous, and heterogeneous** (different languages, ages, interests, etc.).	- Radio and television broadcasts - Newspapers and magazines - Films and advertisements - Online news portals and social media campaigns

36.3 Key Characteristics of Mass Communication

Characteristic	Explanation
Large and Heterogeneous Audience	The audience is vast, diverse, and may differ in education, culture, and profession.
Institutional Source	The sender is usually an **organization or institution** (e.g., Ministry of Information & Broadcasting, Times of India, IGNOU).
Technological Mechanism	Uses **tools and technology** like printing presses, broadcast studios, transmitters, and the Internet for mass reproduction and dissemination.

37. Technology for People with Disabilities

37.1 Introduction

The most meaningful use of technology is to **make life easier for people**—especially for those with **disabilities**.
In developing countries, millions of individuals need **specialized communication and educational tools** to help them participate fully in society.

Modern **Assistive Technologies (AT)** make it possible for people with disabilities to **learn, work, and live independently**, bringing **accessibility and equality** to education and communication.

37.2 Classification of Disabilities

Disabilities are generally classified into **four main groups**:

Type of Disability	Description
1. Visual Impairment	Includes **complete blindness, partial sight, low vision, or colour blindness**.
2. Hearing Impairment	People with **partial or complete hearing loss** who require alternative communication means.
3. Mobility Impairment	Physical limitations due to **paralysis, polio, loss of limbs, or reduced motor control**.
4. Cognitive Impairment	Relates to **memory loss, reduced learning ability, or difficulty processing information** (e.g., dyslexia).

Creating **digital resources** that meet the needs of all these groups is challenging, but possible through **Assistive Technologies**.

37.3 Assistive Technologies (AT)

Definition:
Assistive Technology refers to **any device, software, or system** that helps people with disabilities perform tasks that would otherwise be difficult or impossible.

Purpose:

- Enables **independence** and **self-reliance**
- Promotes **learning and employment**
- Improves **social participation** and **digital inclusion**

37.4 Examples of Assistive Technologies

Assistive Technology	Description / Function	Main Users
Screen Readers	Software that converts text and interface elements into **audio** output.	Visually impaired / blind users
Voice Recognition Software	Allows users to **control computers and input data** using speech.	Users with mobility or typing difficulties
Speech Synthesizers	Convert **written text into spoken words**.	Visually impaired, cognitive impairments
Screen Magnifiers	Enlarge text and graphics on the screen; also modify colour contrast.	People with low vision
Braille Embossers	Produce **Braille printouts** from computer-generated text.	Blind users
Refreshable Braille Displays	Convert digital text into **tactile Braille dots** that can be felt with fingers.	Blind users
Adaptive Keyboards	Specially designed keyboards with **modified key layouts or one-hand use**.	Physically disabled users
Touch Screens / On-screen Keyboards	Enable users to interact with computers using **touch** instead of typing.	Physically disabled users
Reading and Learning Tools	Simplify reading, comprehension, and vocabulary building.	Users with cognitive or learning disabilities

37.5 Matching Assistive Technologies with Disabilities

Type of Disability	Assistive Technologies
Visual Impairment	Screen readers, screen magnifiers, speech synthesizers, refreshable Braille displays, Braille embossers
Hearing Impairment	Generally no specific AT required for computing; however, **captioning, text-based communication**, and **visual alerts** are helpful.
Physical Impairment	Adaptive keyboards, speech recognition systems, touch screens
Cognitive Impairment	Speech synthesizers, learning support software, reading tools, speech recognition systems

37.6 Accessible Website Design

With most educational and government institutions offering online content, **web accessibility** has become essential to ensure that **persons with disabilities can access information**.

However, research shows that only a small number of institutional websites **meet accessibility standards** (e.g., **W3C Web Accessibility Guidelines**).

Freed et al. (2003) outlined **eight guidelines** for accessible digital design:

Guideline	Purpose
1. Provide **text equivalents** for images.	Helps blind users through screen readers.
2. Ensure **access to multimedia** content.	Use captions, transcripts, or sign language.
3. Make **forms accessible** for visually impaired users.	Ensure labels and instructions are screen-reader friendly.
4. Provide **access to data in tables**.	Use proper headers and tags for table navigation.

5. Ensure **access to textbooks** and documents.	Offer digital, large-print, or audio versions.
6. Make **interactive activities** accessible.	Include keyboard navigation and voice control options.
7. Provide **access to graphs and charts**.	Use descriptions or tactile graphics.
8. Ensure **scientific and mathematical expressions** are accessible.	Use markup languages (MathML) or alternative text.

38. Computer Networks

38.1 Definition

A **Computer Network** is a system of **two or more interconnected computers** that can **communicate, share resources, and exchange data**.
Connections may use **cables, telephone lines, radio waves, or satellites**.

38.2 Objectives of Networking

Computer networks are designed to:

- **Share resources** like printers, files, modems, and fax machines.
- **Share application software**, e.g., MS Office.
- **Enable electronic communication** via email, chat, or conferencing.
- **Increase productivity** by allowing multiple users to access and exchange data efficiently.

A complete network includes:

- **Client and server machines** with network operating systems.
- **Transmission media** (wired or wireless).
- **Supporting hardware** such as **routers, bridges, switches, antennas, and towers**.

38.3 Types of Computer Networks

Type	Full Form / Area Covered	Description / Features
LAN	Local Area Network	Covers a **small area** such as a single building or campus. High data transfer speed, low cost, and privately owned.
MAN	Metropolitan Area Network	Covers a **city or town**. Larger than a LAN but smaller than a WAN. Used by government, universities, and large organizations.
WAN	Wide Area Network	Covers **large geographical regions**, even multiple countries. Uses public communication links like telephone lines, satellites, or fiber optics.

38.4 Other Network Characteristics

Category	Meaning	Examples / Notes
Topology	The **physical or logical layout** of connected devices.	Common topologies: **Bus, Star, Ring, Mesh**.
Protocol	A **set of communication rules** that governs data transfer between devices.	Examples: **Ethernet, Token Ring, TCP/IP**.
Architecture	Defines how network components interact.	Two main types: **Peer-to-Peer** and **Client/Server** architectures.

38.5 Network Goals and Motivations

A well-designed network should fulfil several essential goals:

- **Cost Reduction** – Sharing of hardware and software resources.
- **High Reliability** – Backup systems and redundant paths for data.
- **High Throughput** – Efficient transfer of large data volumes.
- **Inter-process Communication** – Smooth data exchange among users.
- **Low Delay** – Quick transmission and response times.
- **Easy Maintenance** – Repairs, upgrades, and expansions with minimal disruption.
- **Interoperability** – Ability to use devices and software from different vendors.
- **Centralised or Distributed Control** – Flexible management of network resources.

38.6 Classification Based on Transmission Technology

Computer networks are also categorized by how they transmit data:

Type	Description / Features
1. Broadcast Networks	- A **single communication channel** shared by all devices. - Each message (packet) contains an **address field** specifying the intended recipient. - All nodes receive the packet, but only the addressed node processes it. - Support **Broadcasting** (to all nodes) and **Multicasting** (to selected groups).
2. Point-to-Point (Switched) Networks	- Consist of **multiple connections between pairs of computers**. - Data travels through intermediate **switching elements**. - Used in most **Wide Area Networks (WANs)**.

38.7 Types of Broadcast Networks

Type	Description
Packet Radio Networks	Use radio waves for data transmission. Suitable for mobile and local communication. Require **radio repeaters** for long distances.
Satellite Networks	Use satellites for global data transmission. Have **long propagation delay** but **large coverage area**.
Local Area Networks (LANs)	Share data among computers within a **limited area** using **Ethernet or Wi-Fi**.

Static Allocation:

- Channel divided into time slots; each machine transmits only in its turn (round-robin method).
- Simple but inefficient when devices have no data to send.

Dynamic Allocation:

- **Centralised:** A controller (e.g., bus arbiter) decides which node transmits next.

- **Decentralised:** Each node decides independently based on internal logic or protocols.

39. Local Area Network (LAN) and Point-to-Point Networks

39.1 Local Area Network (LAN)

Definition:
A **Local Area Network (LAN)** is a computer network that spans a **limited geographical area**, such as a **building, office, or campus**.
LANs are designed for **high-speed data transfer** and **resource sharing** among connected computers.

39.2 Features of LAN

- Confined to **small areas** (typically 5–10 km).
- **High-speed data transmission**, usually between **10 Mbps and 1000 Mbps (1 Gbps)**.
- **Low error rates** and **low cost** setup.
- Allows **data and hardware sharing** among users (e.g., printers, scanners, storage).
- Consists of **personal computers, servers, routers, and switches**.
- Common LAN technology: **Ethernet** (wired) or **Wi-Fi** (wireless).

39.3 Functions and Uses

- **File and resource sharing** (e.g., printers, software, storage).
- **Electronic communication** (e.g., email, chat, file transfer).
- **Centralized management** of user accounts and permissions.
- **Access to shared internet connections**.

39.4 Components of LAN

Component	Function
Nodes (Workstations)	Individual computers connected to the network.
Server	Central computer managing data and network resources.
Network Interface Card (NIC)	Hardware that connects a device to the LAN.
Switch/Hub	Directs data between devices on the network.
Transmission Media	Physical or wireless medium (twisted pair, coaxial, fiber optic, or radio waves).

39.5 LAN Characteristics

Aspect	Details
Topology	Defines physical layout — **Bus, Star, Ring, or Mesh**.
Protocols	Define communication rules — e.g., **Ethernet, Token Ring**.
Media	Can use **wired (cables)** or **wireless (radio signals)**.
Speed	Typically from **10 Mbps to 1 Gbps or higher**.

39.6 Point-to-Point or Switched Networks

Definition:
A **point-to-point network** (also called a **switched network**) connects computers **through dedicated links** or **switching devices**, allowing communication between pairs of machines.

Key Idea:
When data travels from source to destination, it may pass through **one or more intermediate nodes**.
Routing algorithms determine **the best available path**.

Example: International dial-up telephone system.

39.7 Types of Switched Networks

Type	Description / Characteristics	Example
1. Circuit-Switched Networks	- Establishes a **dedicated path** between sender and receiver before communication. - Connection remains active for the entire session (like a phone call). - **Fixed bandwidth** reserved even during silence. - Suitable for **real-time communication**.	Telephone system, ISDN
2. Packet-Switched Networks	- Data is divided into **small packets**, each sent independently with destination addresses. - Packets may take **different routes** and are reassembled at the destination. - **Efficient use of bandwidth**; ideal for digital data transfer. - Suitable for **email, file transfer, and web data**.	Internet, X.25, Frame Relay, ATM, TCP/IP

39.8 Advanced Packet-Switching Technologies

Technology	Description
X.25	Early international standard for packet-switched WANs.
Frame Relay	Designed for **high-speed digital lines**, replacing X.25.
SMDS (Switched Multimegabit Data Service)	Used for **metropolitan networks**.
ATM (Asynchronous Transfer Mode)	Combines packet switching's efficiency with **circuit switching's reliability**; supports **voice, video, and data**.
TCP/IP, IPX/SPX, NetBIOS	Protocols designed to run over packet-switched networks.

39.9 Comparison of Circuit and Packet Switching

Feature	Circuit Switching	Packet Switching
Connection	Dedicated connection established before communication.	No dedicated path; packets are routed individually.
Bandwidth Use	Fixed and continuous.	Shared dynamically among users.
Efficiency	Lower (unused bandwidth wasted).	Higher (no idle channel time).
Delay	Low, fixed delay once connection established.	Variable delay due to dynamic routing.
Use Case	Voice calls, video conferencing.	Internet data, email, file transfer.

40. Network Topology

40.1 Meaning of Network Topology

Network topology refers to the **shape or layout of a network** — the way computers, cables, and other devices are physically or logically arranged and connected.
It determines **how data travels** between devices and **how efficiently communication** takes place.

Topologies are of two types:

- **Physical topology** – the actual physical layout of cables and devices.
- **Logical topology** – the way data flows within the network regardless of physical layout.

When choosing a topology, important factors include:

- Ease of installation and setup
- Flexibility and scalability
- Ease of troubleshooting and maintenance

The most common network topologies are:
Bus, Star, Ring, Tree, Mesh, and Cellular.

40.2 Bus Topology

In **bus topology**, all devices are connected to a single central cable known as the **bus** or **backbone**.
Each workstation is connected in sequence along this cable.
When one computer sends data, all others receive it, but only the intended device accepts the message.

This design is simple and inexpensive, making it suitable for **small networks**.
However, a **fault in the main cable** can bring down the entire network.
Bus topology is most commonly seen in **Ethernet** networks.

Advantages:

- Easy and cheap to install
- Requires less cabling
- Simple to understand and extend

Disadvantages:

- Poor performance as the number of devices increases
- Difficult to identify faults
- A break in the bus cable stops all communication

40.3 Star Topology

In **star topology**, all devices are connected to a **central hub or switch**.
Each device has its own cable to the hub, which manages communication and routes data.
If one device fails, it doesn't affect others—but if the **hub fails**, the entire network stops.

This is one of the most widely used topologies in offices and organizations.

Advantages:

- Easy to install and expand
- Faults can be easily located
- A single device failure doesn't affect the rest of the network

Disadvantages:

- Failure of the central hub shuts down the entire network

- Requires more cabling than bus topology

40.4 Ring Topology

In **ring topology**, each computer is connected to two other computers, forming a **closed loop** or ring. Data travels in one direction, passing through each device until it reaches its destination. Each workstation acts as a repeater to keep the signal strong.

Used often in **Token Ring** networks, this topology is orderly but sensitive to faults.

Advantages:

- Easy to install and expand
- Equal access for all devices (no data collisions)
- Fault isolation is easier than in bus topology

Disadvantages:

- Adding or removing devices disrupts the network
- A single break in the ring can stop communication
- More expensive than bus topology

40.5 Tree Topology

Tree topology is a **hybrid structure** that combines elements of both **star** and **bus** topologies.
It has a **root node (main hub)** connected to one or more **secondary hubs**, which in turn connect to computers or smaller networks.
The layout resembles a **tree with branches**.

It is used in large networks like university campuses or corporate offices.

Advantages:

- Easy to expand and manage
- Faults can be traced easily
- Supports different types of cables and devices

Disadvantages:

- Failure of the main hub affects the whole network
- Requires more cabling compared to simpler topologies

40.6 Mesh Topology

In **mesh topology**, every device is connected directly to every other device in the network.
This creates **multiple paths** for data to travel, ensuring high reliability and redundancy.
If one link fails, data can still be sent through alternate routes.

Mesh networks are mainly used in **critical systems** such as telephone exchanges and data centers.

Advantages:

- Very reliable and robust

- Network continues to function even if one link fails
- Easy fault isolation and high data privacy

Disadvantages:

- Very high cabling and hardware costs
- Complex to install and maintain

40.7 Cellular Topology

Cellular topology is used in **wireless networks**.
The service area is divided into **cells**, each with its own **hub or base station**.
Devices within a cell communicate through their local hub, and hubs are interconnected to form a larger network.

This topology is commonly used in **mobile communication systems** and **Wi-Fi networks**.

Advantages:

- Easy to troubleshoot within a cell
- Flexible and supports wireless communication
- Faults can be isolated at the hub level

Disadvantages:

- Hub-to-hub fault tracking can be complex
- Requires strong coordination among hubs

41. Networking Model

41.1 Concept of a Networking Model

A **networking model** is a structured framework that defines how data moves from one computer to another across a network.
Most networks today are designed using a **layered architecture**, in which each layer handles a specific part of the communication process.

The main purpose of dividing the network into layers is to:

- **Simplify system design and troubleshooting**
- **Separate responsibilities** among layers
- **Enable scalability** so the model can fit small or large systems
- **Allow easy updates or replacements** of one layer without affecting others

Each layer provides **specific services** to the layer above it and communicates through well-defined **interfaces**.
In actual data transfer, information flows **downward through the layers** of the sender, moves through the **physical medium**, and then **upward through the layers** of the receiver.

Communication between layers on different computers is logical, not physical—meaning **layer 3 on one machine interacts conceptually with layer 3 on another**, even though the data physically passes through all lower layers.

41.2 Importance of Layered Design

A layered design brings several advantages:

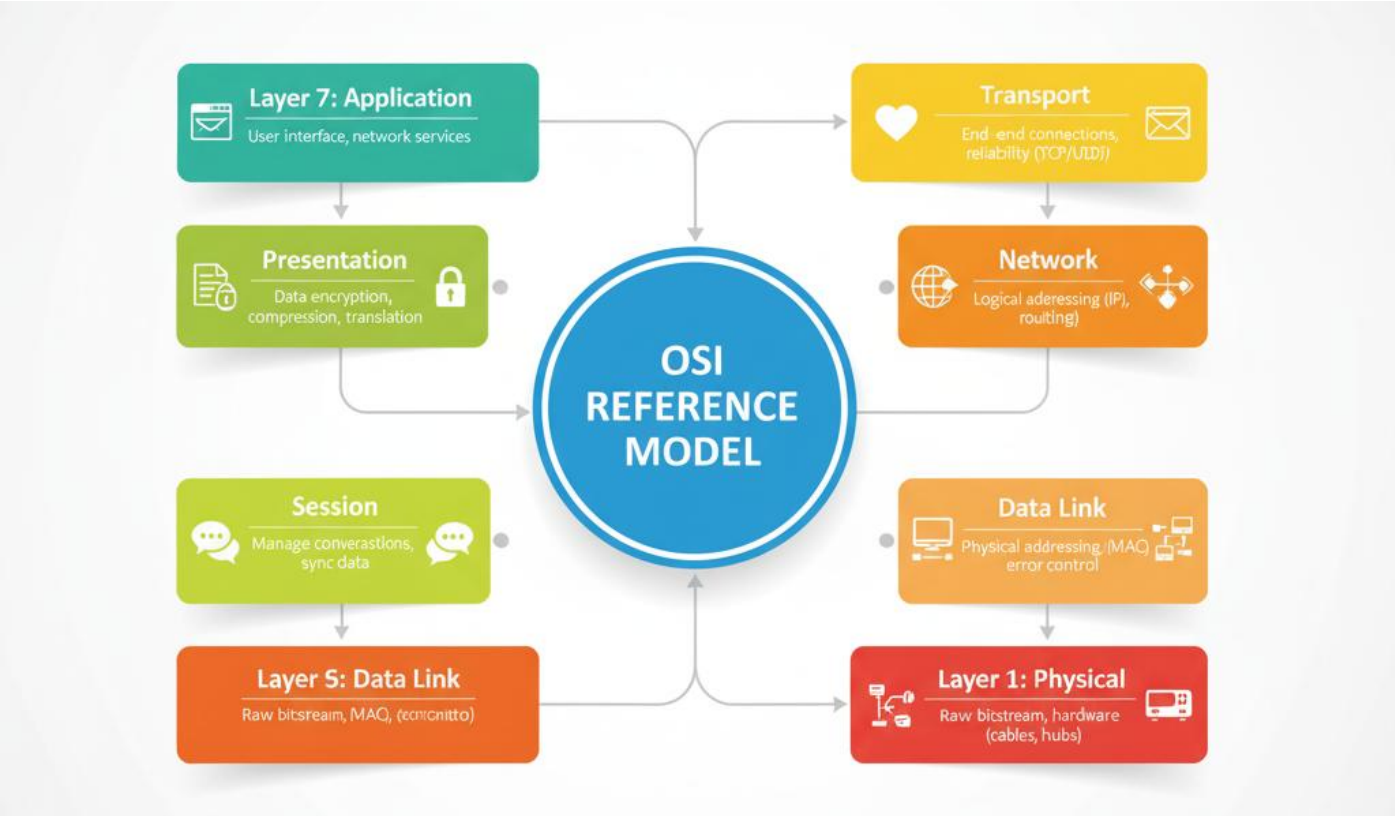
- **Simplifies complexity** by dividing work into smaller, manageable parts
- **Allows interoperability** between different systems and vendors
- **Provides flexibility** to change or improve individual layers
- **Defines clear communication protocols** between layers

Between two adjacent layers lies an **interface**, which defines how data and control information are passed. A well-designed interface ensures that any layer can be modified or replaced without affecting others.

41.3 OSI Reference Model

The **Open Systems Interconnection (OSI) Reference Model**, developed by the **International Standards Organization (ISO)** in **1984**, is a conceptual framework that standardizes the functions of a communication system.

It divides the entire process of data communication into **seven layers**, each responsible for specific tasks. These layers work together to move data from an **application on one computer** to an **application on another** across a network.



OSI Reference Model

The Seven Layers of the OSI Model

Layer 7 — Application Layer

- Closest to the end user.
- Provides services for network applications like email, web browsing, and file transfer.

- Examples: HTTP, FTP, SMTP, DNS.

Layer 6 — Presentation Layer

- Ensures data is in a usable format and understandable by both systems.
- Handles data translation, encryption/decryption, and compression.
- Example: Converts EBCDIC to ASCII or encrypts sensitive data before transmission.

Layer 5 — Session Layer

- Manages sessions or dialogues between communicating systems.
- Establishes, maintains, and terminates connections.
- Handles session checkpoints and recovery.
- Example: Remote procedure calls, session authentication.

Layer 4 — Transport Layer

- Provides **reliable end-to-end delivery** of data between hosts.
- Handles error detection, correction, and flow control.
- Examples: TCP (reliable), UDP (unreliable).

Layer 3 — Network Layer

- Responsible for **logical addressing** and **path selection (routing)**.
- Determines how data is sent from one network to another.
- Examples: IP (Internet Protocol), ICMP, IPX.

Layer 2 — Data Link Layer

- Provides reliable communication between directly connected devices.
- Organizes bits into **frames**, manages **MAC addressing**, and controls access to the physical medium.
- Examples: Ethernet, PPP, HDLC.

Layer 1 — Physical Layer

- Deals with the **hardware transmission** of raw bits over a medium.
- Defines cable types, connectors, voltages, and data rates.
- Examples: Coaxial cables, fiber optics, network interface cards (NICs).

41.4 Grouping of Layers

The seven layers can be broadly categorized into two functional groups:

1. Communication-Oriented Layers (Lower Layers):

- Physical, Data Link, Network, and Transport layers
- Handle data transmission and networking operations

2. Data-Processing-Oriented Layers (Upper Layers):

- Session, Presentation, and Application layers
- Manage software interaction and user communication

42. OSI Protocols and Working of OSI Model

42.1 Meaning of OSI Protocols

The **OSI model** provides a *conceptual framework* for communication between computers, but it does not itself perform communication.
Actual data exchange between systems is made possible through **protocols** — formal sets of rules that govern how data is transmitted, received, and processed across a network.

A **protocol** defines:

- The format of messages
- The order of message exchange
- The actions taken when errors occur

Each protocol performs the functions of **one or more OSI layers**.

42.2 Categories of OSI Protocols

Communication protocols are generally grouped into four main categories:

1. **LAN Protocols**
    - Operate mainly at the **Data Link** and **Network layers**.
    - Define communication across local area networks.
    - Examples: **Ethernet, Token Ring, FDDI**.
2. **WAN Protocols**
    - Operate at the **Physical**, **Data Link**, and **Network layers**.
    - Used for long-distance communication between networks.
    - Examples: **Frame Relay, ATM, X.25, PPP**.
3. **Routing Protocols**
    - Function at the **Network layer**.
    - Determine the best path for data to travel and manage traffic flow.
    - Examples: **RIP, OSPF, BGP, IGRP**.
4. **Network (Upper-Layer) Protocols**
    - Operate at the **Transport**, **Session**, **Presentation**, and **Application layers**.
    - Responsible for user communication and data exchange.
    - Examples: **TCP, UDP, HTTP, FTP, SMTP, SNMP**.

42.3 Working of the OSI Model

Data transfer between two computers follows a **layer-by-layer process**, starting from the application of the sender to the application of the receiver.

Process on the sender's side (System A):

1. The **Application layer** prepares the data for transmission.
2. The **Presentation layer** formats, encrypts, or compresses it if required.
3. The **Session layer** establishes, maintains, and controls the connection between systems.
4. The **Transport layer** breaks the data into smaller **segments**, adds error-checking and flow control.
5. The **Network layer** determines the **best route** for transmission and may further divide the data into **packets**.
6. The **Data Link layer** packages packets into **frames**, adds **source and destination addresses**, and error detection codes.
7. The **Physical layer** converts these frames into **binary bits** and transmits them through the medium (cable or wireless).

Process on the receiver's side (System B):

1. The **Physical layer** receives the raw bits and passes them upward.
2. The **Data Link layer** checks for errors and extracts frames.
3. The **Network layer** reassembles packets and confirms the correct route.
4. The **Transport layer** reassembles the original message from segments.
5. The **Session layer** confirms completion of transfer and closes the connection.
6. The **Presentation layer** decodes, decrypts, or decompresses data.
7. The **Application layer** delivers the final data to the user's program.

This entire process ensures that communication is **reliable, structured, and layer-independent**.

42.4 Layer Interaction

Each layer in the OSI model communicates with:

- The **layer directly above** (service user)
- The **layer directly below** (service provider)
- Its **peer layer** in the other system

For example, the **Data Link layer** in System A communicates with:

- The **Network layer** (above it)
- The **Physical layer** (below it)
- The **Data Link layer** in System B

This structured interaction ensures that every part of the communication process is handled systematically.

42.5 Services Provided by OSI Layers

Each OSI layer provides specific **services** to the layer above it through a defined interface.
There are three basic elements involved in providing these services:

1. **Service User** – The OSI layer that *requests* a service from another layer.
2. **Service Provider** – The OSI layer that *offers* the requested service.
3. **Service Access Point (SAP)** – The *logical point of contact* where one layer can access the services of another.

This service-based design ensures that each layer can evolve independently while maintaining compatibility with others.

43. TCP/IP Reference Model

43.1 Introduction

TCP/IP stands for **Transmission Control Protocol / Internet Protocol**.
It is a **protocol suite** — a collection of communication protocols that work together to enable computers to exchange data over networks like the Internet.

Developed in the **early 1980s** by the **U.S. Department of Defense (DoD)** for the **ARPANET** project, TCP/IP became the foundation for today's Internet.
It was designed as an **open, robust, and platform-independent system**, allowing computers of all types to communicate using a **common language**.

TCP/IP is a **layered model**, much like the OSI model, where each layer performs specific functions in data transmission. However, it uses **fewer layers** (four instead of seven).

43.2 Layers of the TCP/IP Reference Model

The TCP/IP model has **four layers**:

1. **Application Layer**
2. **Transport Layer**
3. **Internet Layer**
4. **Link/Physical Layer**

Each layer interacts with the layer above and below it and provides defined services to the next level.

43.3 Application Layer (Layer 4)

- This is the **top layer** of the TCP/IP model.
- It handles **user-level communication** and includes all protocols and services that applications use to communicate over the network.
- Combines the functions of **Application**, **Presentation**, and **Session layers** of the OSI model.

Main functions:

- Data generation for transmission
- File transfer, email, web access, name resolution, and network management

Common Protocols:

- **HTTP (HyperText Transfer Protocol):** Transfers web pages.
- **FTP (File Transfer Protocol):** Interactive file transfer between systems.
- **SMTP (Simple Mail Transfer Protocol):** Sends email messages.
- **Telnet:** Provides remote login to other systems.
- **DNS (Domain Name System):** Translates hostnames into IP addresses.
- **SNMP (Simple Network Management Protocol):** Monitors and manages network devices.

Interfaces:

- **Windows Sockets (Winsock)** and **NetBIOS** are common interfaces that allow applications to use TCP/IP services.

43.4 Transport Layer (Layer 3)

- Provides **end-to-end communication** between devices.
- Ensures **reliable data transfer**, error detection, and flow control.
- Corresponds to the **Transport layer** and parts of the **Session layer** in OSI.

Core Protocols:

1. **TCP (Transmission Control Protocol):**
    - Connection-oriented and reliable.
    - Ensures that all data packets are delivered in order and without errors.
    - Handles packet sequencing, acknowledgments, and retransmission of lost packets.
2. **UDP (User Datagram Protocol):**
    - Connectionless and unreliable but faster.
    - Used for small data transfers and real-time applications (voice/video).
    - Example uses: DNS, streaming, VoIP.

Responsibilities:

- Establishing and terminating connections
- Segmenting and reassembling data
- Managing data flow and reliability

43.5 Internet Layer (Layer 2)

- Also called the **Network Layer** of TCP/IP.
- Manages the movement of data packets between networks (routing).
- Provides a **connectionless**, **best-effort** delivery service.
- Determines the **path** data takes from source to destination.

Core Protocols:

1. **IP (Internet Protocol):**
    - Defines IP addressing and packet routing.
    - Handles fragmentation and reassembly of packets.
2. **ARP (Address Resolution Protocol):**
    - Maps IP addresses to physical (MAC) addresses on a network.
3. **ICMP (Internet Control Message Protocol):**
    - Reports errors and provides diagnostic information (used by the "ping" command).
4. **IGMP (Internet Group Management Protocol):**
    - Manages IP multicast groups used for group communication.

Functions:

- Logical addressing using IP
- Routing of packets across multiple networks
- Error control through ICMP
- Handling congestion and packet loss

43.6 Link / Physical Layer (Layer 1)

Also known as the **Network Access Layer**.
It is responsible for **placing data packets onto the physical network medium** and **receiving them** from it.

Main functions:

- Defines how data is physically transmitted — voltages, signals, connectors, and media.
- Handles **framing**, **error detection**, and **hardware addressing (MAC)**.
- Supports both **LAN** and **WAN** technologies.

Examples of technologies:

- **Ethernet**, **Wi-Fi**, **Token Ring** (for LANs)
- **Frame Relay**, **ATM**, **X.25** (for WANs)

TCP/IP is independent of any one network type — meaning it can run over **wired, wireless, or satellite links**.

43.7 Relation Between TCP and IP

- **TCP (Transport layer)** provides reliable communication between hosts.
- **IP (Internet layer)** provides addressing and routing between networks.

Together, they ensure that data is properly **packaged, addressed, transmitted, routed, and received** across complex networks.

- TCP ensures the **data arrives correctly**.

- IP ensures it **reaches the right destination**.

44. Network Architecture

44.1 Meaning

Network Architecture refers to the design framework that defines how computers and devices within a network are structured, connected, and share resources. It specifies how data flows, how tasks are divided among systems, and how communication takes place.

Based on architecture, computer networks are broadly classified into two main types:

1. **Client/Server Architecture**
2. **Peer-to-Peer Architecture**

44.2 Client/Server Architecture

In **Client/Server Architecture**, the network is organized around two types of devices — **clients** and **servers**.

- The **client** is a personal computer or workstation that **requests services or resources**.
- The **server** is a powerful computer that **provides services**, manages shared resources, and responds to client requests.

These systems are connected through a **Local Area Network (LAN)** or **Wide Area Network (WAN)**.

In this model, the **client handles the user interface** and sometimes part of the processing, while the **server manages databases, applications, or files**.
This structure is also called a **centralized or server-based network**, as all major operations depend on one or more servers.

Main Components

- **Database Server:** Stores and manages data, processes SQL requests from clients.
- **Application Server:** Performs business logic or application-level processing.
- **Network Operating System (NOS):** Controls data exchange, security, and access rights.
- **DBMS and Transaction Monitor:** Maintain data integrity and handle concurrent user access.

Forms of Client/Server Model

1. **Non-Client/Server:**
   The server acts as a **remote disk storage**, and all processing occurs at the client side.
   This can overload the network as large amounts of data are constantly transferred.
2. **Two-Tier Client/Server:**
   The **client sends SQL requests** to the server. The **server processes and returns only relevant records**, reducing network traffic.
   This is the **most common** and basic form of client/server architecture.
3. **Three-Tier Client/Server:**
   Introduces an **intermediate layer (application server)** between client and database server.
   - The client handles presentation.

- The application server processes business logic.
- The database server manages data.
  This structure improves **security**, **scalability**, and **maintenance**.

Advantages

- Centralized control of data and security
- High performance and reliability
- Easy to upgrade or maintain
- Efficient resource utilization

Limitations

- Costly to implement and manage
- Network failure can disrupt all clients
- Requires skilled administration

44.3 Peer-to-Peer (P2P) Architecture

In a **Peer-to-Peer Network**, each workstation functions as **both a client and a server**.
There is **no central server** — every computer can share files, devices, and data directly with others.

All computers have **equal status**, and data or resources are **distributed** throughout the network. Users control access to their own resources, such as folders or printers.

P2P networks are generally used in **small offices, home networks, or temporary setups**, where cost and simplicity are priorities.

Characteristics

- Each node (computer) has **equal authority**.
- **No centralized administration** or server.
- Resources and data are **shared voluntarily** by users.
- Suitable for networks with fewer than 10–15 computers.

Advantages

- Simple and inexpensive setup
- No need for a dedicated server
- Easy to install and maintain

Limitations

- Difficult to secure and manage
- Performance drops under heavy load
- No centralized backup or control
- Data consistency can be hard to maintain

44.4 Key Difference

- **Client/Server** networks have a **dedicated central server**; **Peer-to-Peer** networks do not.
- **Client/Server** is more suitable for **large organizations**, while **Peer-to-Peer** fits **small or temporary networks**.
- **Client/Server** networks provide **better security, reliability, and scalability**, while **P2P** offers **low cost and simplicity**.

45. Popular Networks

As computers became central to every field, different types of networks emerged — **public networks, research networks, cooperative networks, and corporate networks**.
These differ in their **administration, design, facilities, and user communities**.
Among the most significant are **Novell NetWare** and **ARPANET**, both of which shaped modern networking systems.

45.1 Novell NetWare

Novell NetWare was one of the most widely used **PC-based networking systems** in the 1980s and 1990s.
It was developed by **Novell Inc.** and based on the **Client/Server model**, where one or more computers act as servers and others as clients.
NetWare allowed different types of PCs and devices to communicate efficiently within a network.

Versions

- **NetWare 2.x (286)** – for Intel 286-based systems
- **NetWare 3.x (386)** – for 386 and 486 systems
- **NetWare 4.x** – compatible with most modern PCs and network types

Architecture and Functioning

In a NetWare network:

- The **server** runs the NetWare operating system and manages shared data and services.
- **Clients** (workstations) send requests to the server for files, printers, or databases.
- The network runs over **LANs or WANs** using Ethernet, Token Ring, or ARCnet.

NetWare uses a proprietary protocol stack based on the **Xerox Network System (XNS)**, but heavily modified.

Types of File Servers

1. **Dedicated File Server:**
   Runs only NetWare; used in large networks for efficiency and higher security.
2. **Non-Dedicated File Server:**
   Runs both NetWare and user applications; suitable for small networks where the server can also act as a client.

Other Servers in NetWare

- **Print Server:** Manages print jobs and shared printers.

- **Message Server:** Handles user emails and message exchange.
- **Database Server:** Manages database records and queries for clients.

Protocols in NetWare

- **IPX (Internet Packet Exchange):** A connectionless internetwork protocol similar to IP but using 12-byte addresses.
- **SPX (Sequenced Packet Exchange):** A transport protocol ensuring reliable communication.
- **NCP (Network Core Protocol):** The main service protocol responsible for data transport, authentication, and file access.
- **SAP (Service Advertising Protocol):** Allows servers to advertise their services across the network.

NetWare's layered model omitted the Session and Presentation layers, combining most functions into Application, Transport, and Network layers.

Importance

Novell NetWare set early standards for **file sharing, network security, and resource management** in local networks. It dominated business networking before TCP/IP and Windows Server technologies became widespread.

45.2 ARPANET (Advanced Research Projects Agency Network)

ARPANET was the **first operational packet-switching network** and the **foundation of the modern Internet**.
It was developed in **1969** by the **U.S. Department of Defense's ARPA (Advanced Research Projects Agency)** to create a **robust communication system** that could survive partial network failures — a key concern during the Cold War.

Purpose

The main goal of ARPANET was to:

- Enable secure, decentralized communication between military and research computers.
- Support data sharing among academic and government research institutions.
- Develop networking technologies that could continue functioning even if parts of the system were destroyed.

Development and Growth

- **1969:** ARPANET connected **four universities** — UCLA, Stanford Research Institute, UC Santa Barbara, and the University of Utah.
- **1972:** Expanded to **37 host computers**.
- **1973:** Established **international connections** with England and Norway, making it the first global research network.
- **1983:** Adopted **TCP/IP** as the standard communication protocol for all connected computers.
- **1990:** Officially **retired**, replaced by the faster **NSFNET (National Science Foundation Network)**, which evolved into today's Internet.

Contributions

- Introduced the **packet-switching technology**, where data is broken into packets that travel independently.
- Developed **TCP/IP protocols**, which remain the foundation of global Internet communication.
- Pioneered the concept of **network resilience** — a system that continues functioning despite local failures.
- Led to the creation of **MILNET**, a separate network for military communication.

Legacy

ARPANET laid the groundwork for the **Internet**, allowing smaller networks and universities to interconnect into a "**network of networks**."
Its success inspired the creation of global research and commercial networks, transforming how data is shared and accessed worldwide.

46. The Internet

46.1 Introduction

The **Internet** is the **world's largest computer network**, often described as a **network of networks**.
It connects millions of computers worldwide, allowing users to **communicate, share data, and access information** regardless of location.

The Internet originated from the **interconnection of ARPANET and NSFNET** in the early 1980s.
Its "official birth" is considered around **1982**, when different networks (like BITNET and EARN) adopted the **TCP/IP protocol** as a common standard, unifying previously fragmented systems.

The term **Internet** comes from the words *"interconnection"* and *"network."*

46.2 Development and Coordination

The development of the Internet is managed and guided by several international organizations:

- **ISOC (Internet Society):**
  A non-profit body that promotes Internet use globally, supports developing nations, and maintains usage statistics.
- **IAB (Internet Architecture Board):**
  Plans long-term development, oversees Internet architecture, and maintains the **RFC (Request for Comments)** documents that record technical standards and protocols.
- **IETF (Internet Engineering Task Force):**
  A working group under ISOC responsible for developing and updating Internet standards, including TCP/IP and security protocols.

Together, these bodies ensure the Internet remains open, interoperable, and scalable.

46.3 Reasons for Rapid Growth

Several key factors contributed to the explosive growth of the Internet:

1. **User-friendly browsers** — made navigation simple for non-technical users.
2. **Improved telecommunications** — faster and cheaper global connectivity.

3. **Expansion of electronic services** — email, online banking, e-commerce, and data sharing.
4. **Information Superhighway projects** — government and corporate investments in global connectivity.

46.4 Protocol Foundation

All Internet communication relies on the **TCP/IP protocol suite**, which ensures interoperability among devices and networks worldwide.

- **IP (Internet Protocol):**
  Provides a **unique 32-bit address** (IP address) to every computer and handles routing of data packets.
- **TCP (Transmission Control Protocol):**
  A **connection-oriented** protocol that ensures reliable delivery and correct sequencing of data. It also manages **port numbers** for various services within a system.

Together, TCP/IP allows computers of all types and operating systems to communicate seamlessly.

46.5 Major Internet Services

The Internet provides a wide range of services. Some of the most important include:

- **Email (SMTP):** Electronic mail transmission.
- **File Transfer (FTP):** Uploading and downloading files between computers.
- **Remote Login (TELNET):** Accessing and controlling remote computers.
- **Menu-Based Access (GOPHER):** Early method of structured information retrieval.
- **Information Retrieval (WAIS / Z39.50):** Searching across distributed databases.
- **World Wide Web (HTTP):** Accessing hyperlinked documents and multimedia content.
- **Network Testing (PING):** Checking connectivity between computers.

46.6 Internet Connectivity Methods

There are three main ways to connect a computer to the Internet:

1. **Dial-Up Access:**
   - Uses a **modem and telephone line** to connect temporarily to an Internet Service Provider (ISP).
   - Provides low speeds (up to 56 Kbps).
   - Suitable for basic communication or text-based browsing.
2. **Dial-Up Networking:**
   - Similar to dial-up but uses a local network setup (LAN) that connects through one central Internet connection.
3. **Leased Lines:**
   - A **permanent, high-speed connection** rented from a telecom provider.
   - Provides continuous data transmission and higher reliability.

- Ideal for organizations requiring large-scale or constant data access.

46.7 Working of the Web

The **World Wide Web (WWW)** operates on a **client-server model**:

- The **client** is the user's computer, running a **web browser** (such as Chrome, Firefox, or Edge).
- The **server** hosts web pages and responds to browser requests.

How it works:

1. The user enters a **URL (Uniform Resource Locator)** in the browser.
2. The browser sends a request to the web server using **HTTP (HyperText Transfer Protocol)**.
3. The server sends back the requested data.
4. The browser interprets the data (written in **HTML – Hypertext Markup Language**) and displays it to the user.

The **Web** is held together by **hypertext** and **hyperlinks**, allowing users to navigate easily between related pages — a process commonly called **browsing** or **surfing the Internet**.